Capstone Project – Final Report

Automatic Ticket Assignment(NLP)

Batch - **DEC-A G: 4 (2020-21)**

Milestone–2 Submission

12-12-2021

AIML-Dec “A” G: 4 Team Structure:

Mentor	:	**Sumit Kumar**
Program Coordinator	:	**Shabana Khan, Namrata Singh**

Team Members:

- ✓ **Priya Moily**
- ✓ **Priyanka Gupta**
- ✓ **Priyank Bhuch**
- ✓ **Avinash Balani**

Github link: https://github.com/pbhuch/GL_DecA_G4_NLP1

Table of Contents

Abbreviations and Acronyms:

AIML	Artificial Intelligence Machine Learning
EDA	Exploratory Data Analysis
IP	Internet Protocol
ITSM	Information Technology Service Management
IT	Information Technology
MTTR	Mean Time to Restore
NLP	Natural Language Processing
POS	Parts of Speech

Executive Summary:

This document is the Final report of the Capstone Project- Automatic Ticket Assignment (NLP) for the Batch Dec-A G: 4

The purpose of this Capstone project is to showcase the power of Natural Language Processing (NLP) and how they can complement the matured ITSM processes, especially the Service Desk of an organization in automatically identifying the functional group to assign a service desk incident

This Automated Ticket assignment system will help to increase the service desk efficiency, result in rationalization of the team structure of L1/L2 by enabling them in spending of their time and effort in some other useful work than the mundane assignment of tickets along with ensuring customer satisfaction by getting all the issue resolved with lesser time.

Technologies Used:

- ✓ **Python**: This application is developed using the Python. Natural Language Processing libraries are used extensively in this project.
- ✓ **Google Collaboratory** from Google Research a hosted Jupyter notebook service is used for the python coding and running the code.
- ✓ **Github**, the software development platform has been used here for the collaboration of data, code documentation.
- ✓ **Zoomplatform**isusedforcollaboratingbetweentheteammembers,fordailycalls with screen sharing

1. Summary of problem statement, data and findings:

Summary of Problem Statement:

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.

Current Process:

In the support process, incoming incidents are analyzed and assessed by 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.

Currently the incidents are created by various stakeholders (Business Users, IT Users and Monitoring Tools) within IT Service Management Tool and are assigned to Service Desk teams (L1 / L2 teams). This team will review the incidents for right ticket categorization, priorities and then carry out initial diagnosis to see if they can resolve.

Around ~54% of the incidents are resolved by L1 / L2 teams. Incase L1 / L2 is unable to resolve, they will then escalate / assign the tickets to Functional teams from Applications and Infrastructure (L3 teams). Some portions of incidents are directly assigned to L3 teams by Monitoring tools or Callers / Requestors. L3 teams will carry out detailed diagnosis and resolve the incidents. Around ~56% of

Incidents are resolved by Functional / L3 teams. Incase if vendor support is needed, they will reach out for their support towards incident closure.

L1 / L2 needs to spend time reviewing Standard Operating Procedures (SOPs) before assigning to Functional teams (Minimum ~25-30% of incidents needs to be reviewed for SOPs before ticket assignment). 15 min is being spent for SOP review for each incident. Minimum of ~1 FTE effort needed only for incident assignment to L3 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. Additional effort needed for Functional teams to re-assign to right functional groups. During this process, some of the incidents are in queue and not addressed timely resulting in poor customer service.

Guided by powerful AI techniques that can classify incidents to right functional groups can help organizations to reduce the resolving time of the issue and can focus on more productive task

Data Set

The data file “input_data.xlsx” is provided as a prerequisite in the capstone project and this file contains the dataset required as input to the project.

The data file (input_data.xlsx) contains four columns as described below

Short description	Description	Caller	Assignment group
login issue	-verified user details.(employee# & manager na	spxjnwir pjlcoqds	GRP_0
outlook	received from: hmjdrvpb.komuaywn@gmail.cor	hmjdrvpb komuaywn	GRP_0
cant log in to vpn	received from: eylqgodm.ybqkwiam@gmail.con	eylqgodm ybqkwiam	GRP_0
unable to access hr_tool page	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0
skype error	skype error	owlgqjme qhcozdfx	GRP_0
unable to log in to engineering tool and skype	unable to log in to engineering tool and skype	eflahbxn ltdgrvkz	GRP_0
event: critical:HostName_221.company.com t	event: critical:HostName_221.company.com the	jyoqwxhz clhxsoqy	GRP_1
ticket_no1550391- employment status - new r	ticket_no1550391- employment status - new no	eqzibjhw ymebpoih	GRP_0
unable to disable add ins on outlook	unable to disable add ins on outlook	mdbegvct dbvichlg	GRP_0
ticket update on inplant_874773	ticket update on inplant_874773	fumkcsji sarmtlhy	GRP_0
engineering tool says not connected and unal	engineering tool says not connected and unable	badgknqs xwelumfz	GRP_0
hr_tool site not loading page correctly	hr_tool site not loading page correctly	dcqsolkx kmsijcuz	GRP_0
unable to login to hr_tool to sgxqsuojr xwbes	unable to login to hr_tool to sgxqsuojr xwbesorf	oblekmrw qltgvspb	GRP_0
user wants to reset the password	user wants to reset the password	iftldbmu fujslwby	GRP_0
unable to open payslips	unable to open payslips	epwyvjsz najukwho	GRP_0
ticket update on inplant_874743	ticket update on inplant_874743	fumkcsji sarmtlhy	GRP_0
unable to login to company vpn	received from: xyz@company.comhi,i am unable	chobktqj qdamxfuc	GRP_0
when undocking pc , screen will not come bac	when undocking pc , screen will not come back	sigfdwcj reofwzlm	GRP_3

DataFindings:

- ✓ Short description & Long description tells us about the type of issue.
- ✓ Observation, 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.
- ✓ Stopwords are present in the description.
- ✓ The Assignment Group column is the target variable and classes among which the incidents will be assigned
- ✓ Caller is the name who has raised the incident
- ✓ The dataset has total of 8500 samples.
- ✓ The 8500 record share distributed across 73 Assignment groups
- ✓ Almost 50% of samples belong to one Assignment Group(Group-0) and this means data imbalance between the class
- ✓ Few of the Assignment groups 5 to1 samples and these groups will impact the performance of the Model big time

- ✓ AsdepictedintheVisualisation-1insection5,the data is skewed and there is high imbalance between assignment groups (target classes).
- ✓ 3975incidentswereassignedtoAssignmentGroup-0andthisismajoritygroup
- ✓ There are 6 groups with only 1 record of data sample
- ✓ There are 5 groups with only 2 incidents assigned.
- ✓ There are 5 groups with only 3 incidents assigned to them.
- ✓ There are 2 groups with only 4 incidents assigned to them.
- ✓ There is only one group with 5 incidents assigned to them.
- ✓ Data has Null values:

Columns	Count of NULL values
Short description	2
Description	1
Assignment group	0
Caller	0

- ✓ Don't see parent/ child relation between any incidents.
- ✓ All incident belong to L1, L2 support (basis data finding)
- ✓ Looking at the description, incidents are raised in different languages.
- ✓ There are some duplicate records identified, detail analysis will be done by performing EDA.
- ✓ There are some Callers who raised same type of incidents During EDA & preprocessing
- ✓ Detail analysis is required & handling of such data.

Implications:

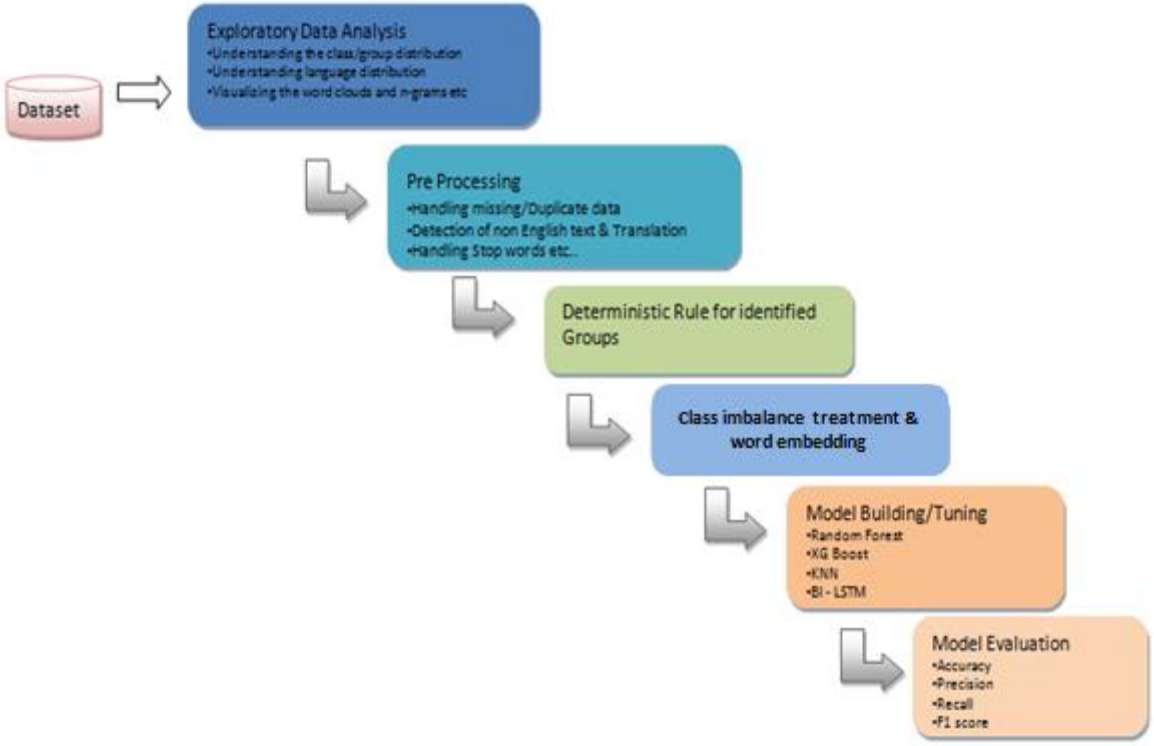
- ✓ DataishighlyimbalancedbetweenGroup-0andRestoftheAssignmentGroups
- ✓ DataimbalancewillbeimpactingtheModelperformanceanditwillbebiasedtowardsmajorityclasses
- ✓ The non-English data have to be either handled with language translation or to be dropped
- ✓ Duplicate data removal & handling of special character.
- ✓ Thehandlingofcallercolumnisbriefedin Pre-processingsection

2. Overview of the Final Process:

Problem Methodology:

The problem statement (incident ticket analysis) is text-based data as input and hence falls in the Natural Language Processing category of AI methodology. Following Flow diagram depicts the overall Pipeline and methodology followed end to end for addressing incident Ticket Assignment problem.

Predominantly there are SIX overall steps in the process and it' san iterative process where the there is always a room to the keep improving the performance of the Model.



Salient Features of the Data,

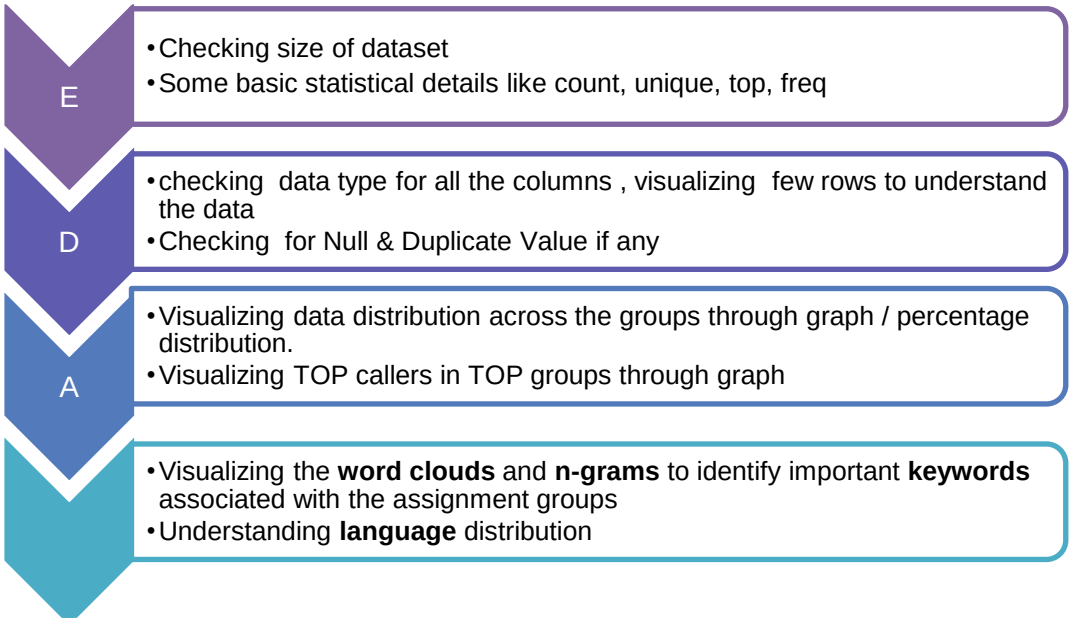
- ✓ Data is highly imbalanced between Group-0 and rest of the Assignment Groups
- ✓ Short Description and Description columns will be making the Corpus of the Model
- ✓ Data seems to be a free text and does not have defined structure or format

- ✓ Similar text content in the incidents is assigned to multiple Assignment Groups in few cases basis same caller has raised those incidents.
- ✓ One of the observations is that Large set of incidents are generated by same Caller
- ✓ The Callers are free to create incidents in any Language and hence we can find roughly 8 to 10% of incidents in languages other than English
- ✓ Large set of incidents are auto generated by the system and do have same text content like Job Scheduler Id, Date, time stamp etc.

EDA (Exploratory data analysis) steps:

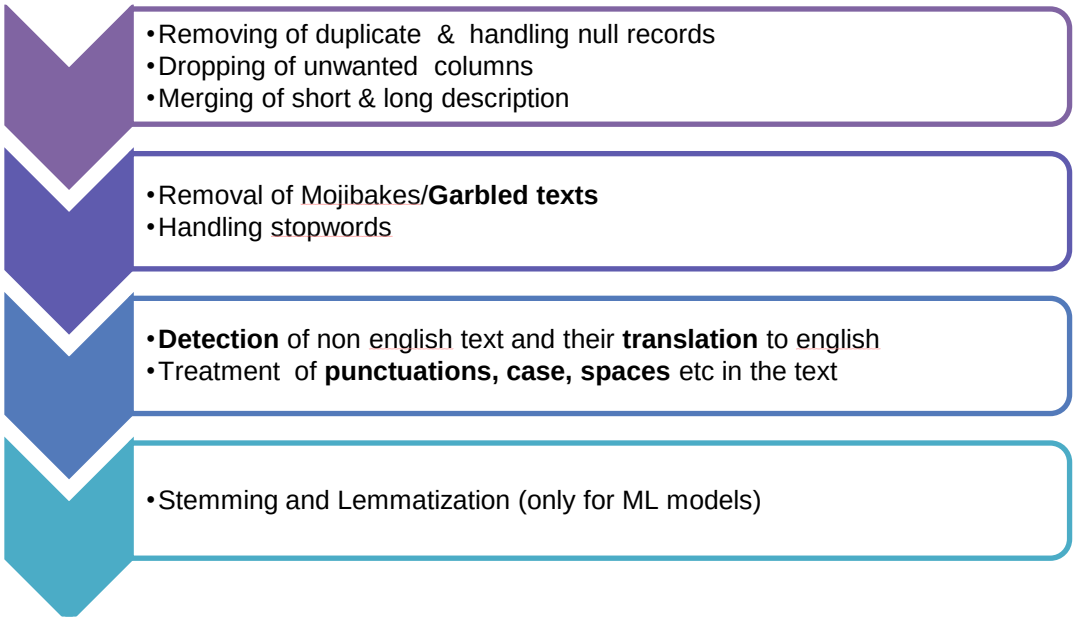
High-level EDA steps mentioned below

Note: Some step in EDA &pre processing could be overlapping, however EDA part is mentioned separately



Data pre-processing steps:

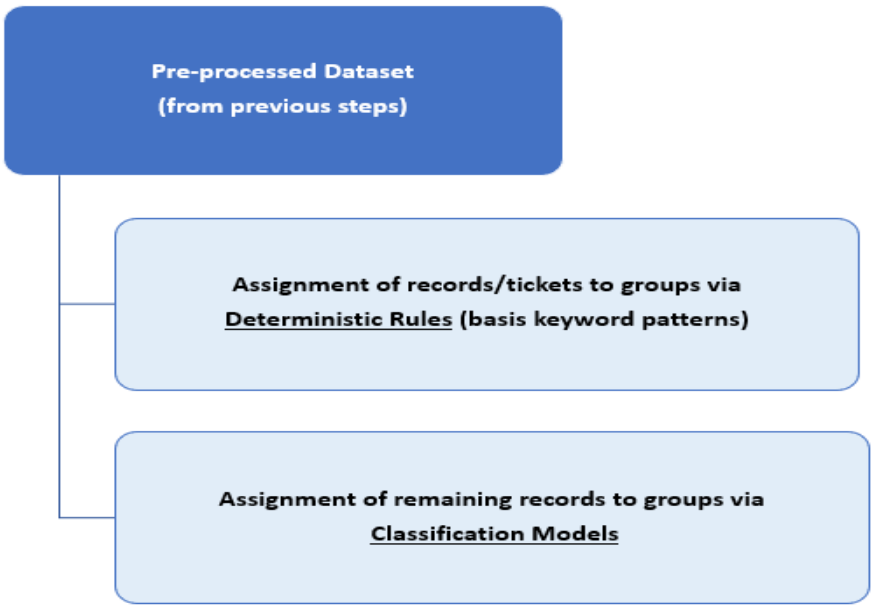
High-level data pre-processing steps followed in our solution



Deterministic Rule

Simple deterministic rules can be applied from processes that are already in place. We have identified a few Keywords which depict incidents/tickets belonging to that specific group

- ✓ **Workflow for Data processing for Rule based & Model building**.
- ✓ By looking at the data though visualization we can see many of the tickets surely belong to specific groups. In such scenarios it is possible to separate known incident from the entire data set.
- ✓ Even some of the group's tickets are similar to that of other & hence we can think of merging the incidents in single groups.



Class Imbalance treatment

- ✓ Treating the class imbalance could be very important as it may have an impact on the performance of our prediction models.
- ✓ We would explore the following methods to treat class imbalance and choose the one which has a better impact on model performance
  - Data augmentation technique
  - Data upsampling through resampling

Different Model building technique

- ✓ NaïveBayes(Multinomial)Classifier
- ✓ Random Forests
- ✓ XGBoost
- ✓ SVC
- ✓ k-NN
- ✓ LSTM & Bidirectional LSTM
- ✓ GRU

Step-by-step walkthrough the solution:

Understanding the problem statement and the business domain

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.

To reduce the manual intervention and its associated problems as highlighted above, the goal here is to build a classifier using ML and Deep Learning models that can automatically classify tickets, by analyzing the text description, to the relevant assignment groups.

Working as a Team

- ✓ Working as a team, there was daily calls scheduled to connect and brainstorm
- ✓ Used Git-hub for tacking our project
- ✓ Define plan with high level activities and due dates
- ✓ As a team, we did a deep dive into the problem statement by sharing each of our experience with incident management process& distributed task to close project on time.

Findings

- ✓ This is very common problem among organization across industries
- ✓ There are multiple levels of parties involved till the ticket is assigned to the final assignment group
- ✓ Incidents are raised either by Service Help Desk personal or Digital Assistant Robot

Understanding the dataset and Data Exploration(EDA)

- ✓ Used excel pivot and features to explore the data and relationships between various columns or features
- ✓ Importing all the relevant libraries to perform EDA

- ✓ Compared Description & Short Description columns to make sure they are not repetitive and contain different text content
- ✓ Analyze the Caller Column to see if any inference with Assignment groups
- ✓ Looked for data in details for presence of URLs, Digits, special characters, new line characters, spaces
- ✓ Analyze the data for multiple language presence and ratio of the sample in English vs other languages
- ✓ Used Google Sheets for identifying the languages
- ✓ Used EDA's like value counts, Group by, various visualizations, word count, outlier with box plots, piecharts to explore the data

Sample data from the dataset

Short description	Description	Caller	Assignment group
login issue	-verified user details.(employee# & manager na	spxjnwir pjlcoqds	GRP_0
outlook	received from: hmjdrvpb.komuaywn@gmail.cor	hmjdrvpb komuaywn	GRP_0
cant log in to vpn	received from: eylqgodm.ybqkwiam@gmail.con	eylqgodm ybqkwiam	GRP_0
unable to access hr_tool page	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0
skype error	skype error	owlgqjme qhcozdfx	GRP_0
unable to log in to engineering tool and skype	unable to log in to engineering tool and skype	eflahbxn ltdgrvkz	GRP_0
event: critical:HostName_221.company.com t	event: critical:HostName_221.company.com the	jyoqwxhz clhxsoqy	GRP_1
ticket_no1550391- employment status - new r	ticket_no1550391- employment status - new nor	eqzibjhw ymebpoih	GRP_0
unable to disable add ins on outlook	unable to disable add ins on outlook	mdbegvct dbvichlg	GRP_0
ticket update on inplant_874773	ticket update on inplant_874773	fumkcsji sarmtlhy	GRP_0
engineering tool says not connected and una	engineering tool says not connected and unable	badgknqs xwelumfz	GRP_0
hr_tool site not loading page correctly	hr_tool site not loading page correctly	dcqsolkx kmsijcuz	GRP_0
unable to login to hr_tool to sgxqsuojr xwbes	unable to login to hr_tool to sgxqsuojr xwbesorf	oblekmrw qltgvspb	GRP_0
user wants to reset the password	user wants to reset the password	iftldbmu fujslwby	GRP_0
unable to open payslips	unable to open payslips	epwyvjsz najukwho	GRP_0
ticket update on inplant_874743	ticket update on inplant_874743	fumkcsji sarmtlhy	GRP_0
unable to login to company vpn	received from: xyz@company.comhi,i am unable	chobktqj qdamxfuc	GRP_0
when undocking pc , screen will not come bac	when undocking pc , screen will not come back	sigfdwcj reofwzlm	GRP_3

The dataset that's provided consists of 8500 records, where each record is assigned to one of the groups among a total of 74 available groups. The dataset contains following 4 columns, which are all string:

Importing Libraries & Data

```
!pip install ftfy
from time import time
from PIL import Image
from zipfile import ZipFile
import os, sys, itertools, re
import tensorflow as tf
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
import plotly as py
import plotly.graph_objs as go
import plotly.express as px
from plotly.offline import init_notebook_mode, iplot, plot
from sklearn.preprocessing import QuantileTransformer
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,f1_score,recall_score,precision_score, confusion_matrix, classification_repor
import sklearn.neighbors._base
```

```
[ ]  dataset = pd.read_excel('/content/sample_data/input_data.xlsx')
     dataset.shape

     (8500, 4)
```

EDA

1. Shape/ size of dataset

```
[ ]  dataset1.head(20)
```

	Short description	Description	Caller	Assignment group
0	login issue	-verified user details.(employee# & manager na...	spxjnwir pjlcoqds	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: eylqgodm.ybqkwiam@gmail...	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
5	unable to log in to engineering tool and skype	unable to log in to engineering tool and skype	eflahbxn ltdgrvkz	GRP_0
6	event: critical:HostName_221.company.com the v...	event: critical:HostName_221.company.com the v...	jyoqwxhz clhxsoqy	GRP_1
7	ticket_no1550391- employment status - new non-...	ticket_no1550391- employment status - new non-...	eqzibjhw ymebpoih	GRP_0
8	unable to disable add ins on outlook	unable to disable add ins on outlook	mdbegvct dbvichlg	GRP_0
9	ticket update on inplant_874773	ticket update on inplant_874773	fumkcsji sarmtlhy	GRP_0

```
dataset = pd.read_excel('/content/sample_data/input_data.xlsx')
dataset.shape
```

```
(8500, 4)
```

2. Checking for Null, Duplicate

```
dataset.isnull().sum()
```

```
Short description    8
Description          1
Caller               0
Assignment group     0
dtype: int64
```

```
duplicate = dataset[dataset.duplicated()]
```

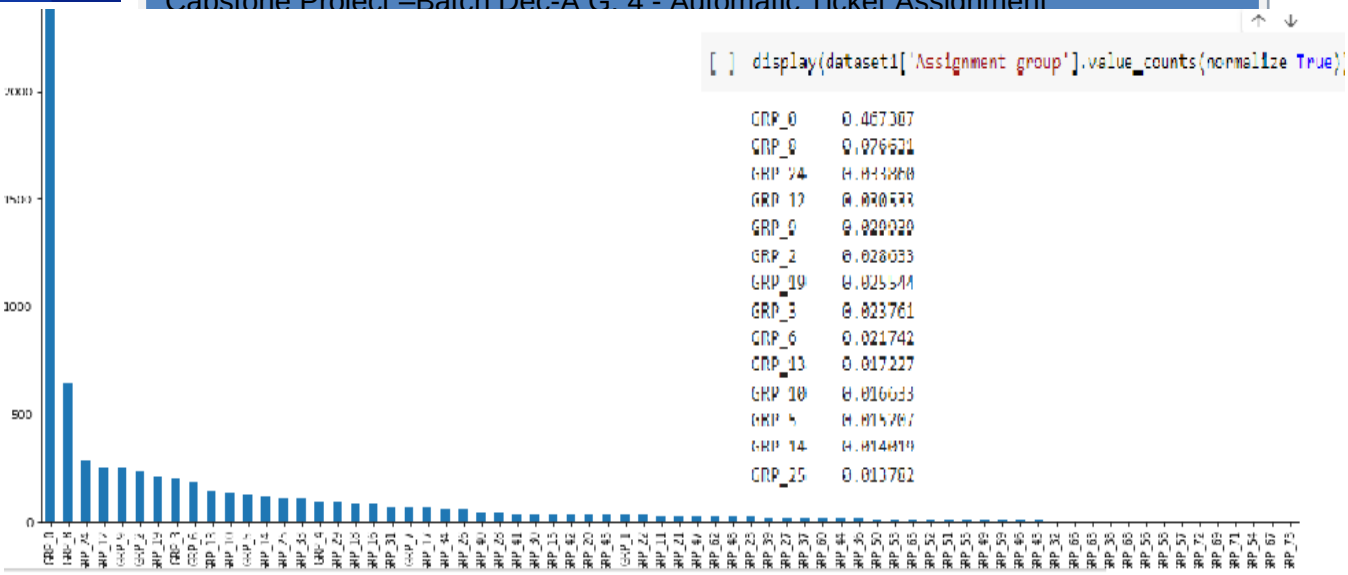
```
duplicate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 83 entries, 51 to 8405
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Short description  83 non-null     object
 1   Description        83 non-null     object
 2   Caller             83 non-null     object
 3   Assignment group   83 non-null     object
dtypes: object(4)
memory usage: 3.2+ KB
```

Null & Duplicate records are identified. Null records is filled with “NaN” & duplicate records are removed

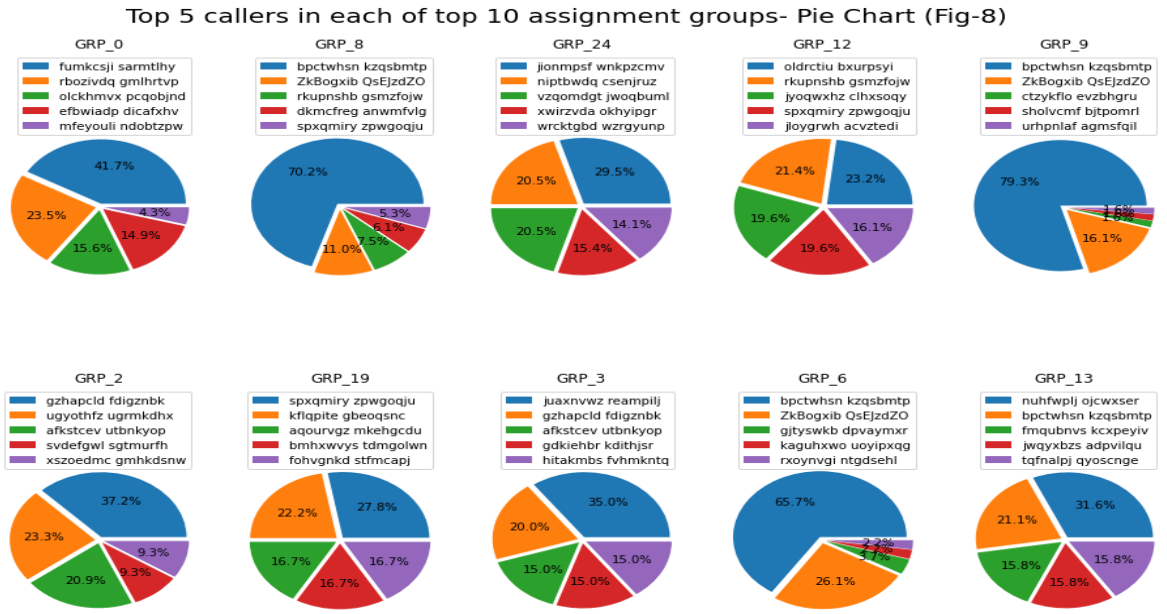
3. Data visualization for checking Data balance

ThedataisveryimbalanceandmajorityofthesamplesareunderGroup-0



- ✓ GRP_0 is the most assigned **group with ~47%** of all tickets assigned to it
- ✓ GRP_8 has second most assignment with 7.7% of tickets assigned to it
- ✓ ~40 groups have just 30 or less tickets assigned amongst which 6 groups have just 1 ticket and 4 groups have just 2 tickets each.
- ✓ This shows that the classes are severely imbalanced, and we would need to treat those ahead

4. Visualizing top callers

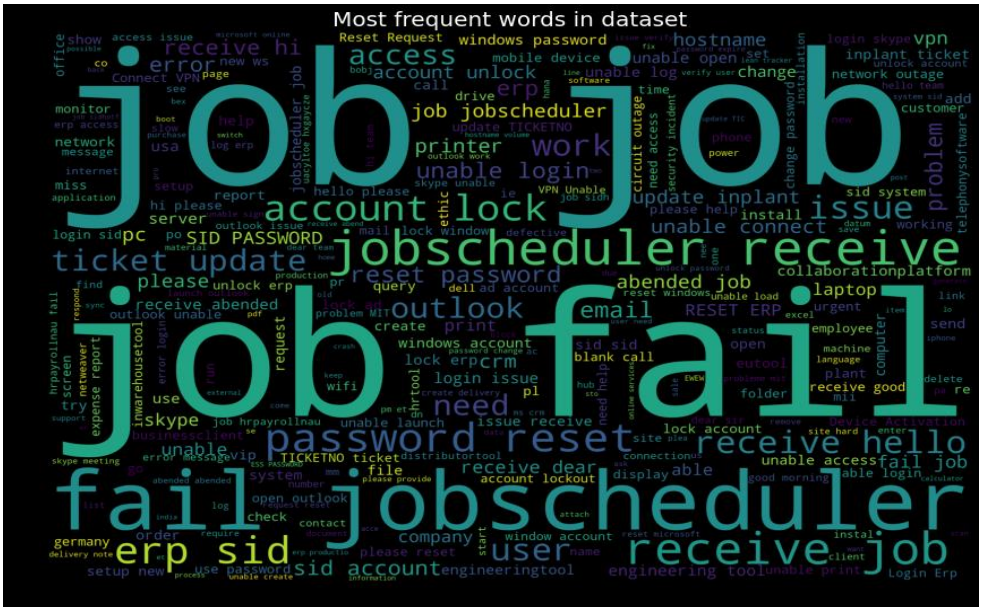


5. Visualizing Word Clouds

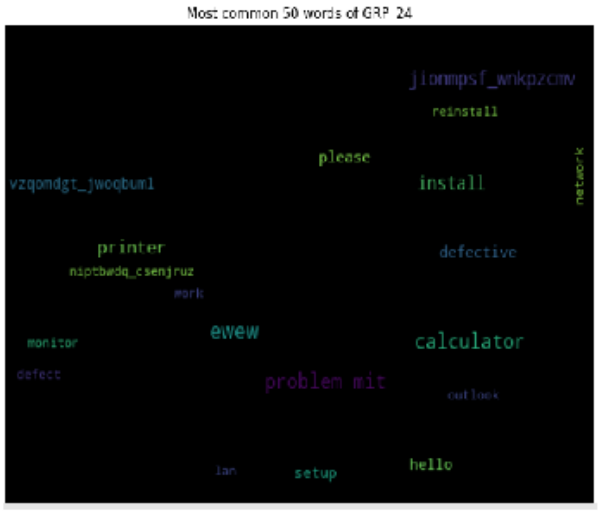
A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is. Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data.

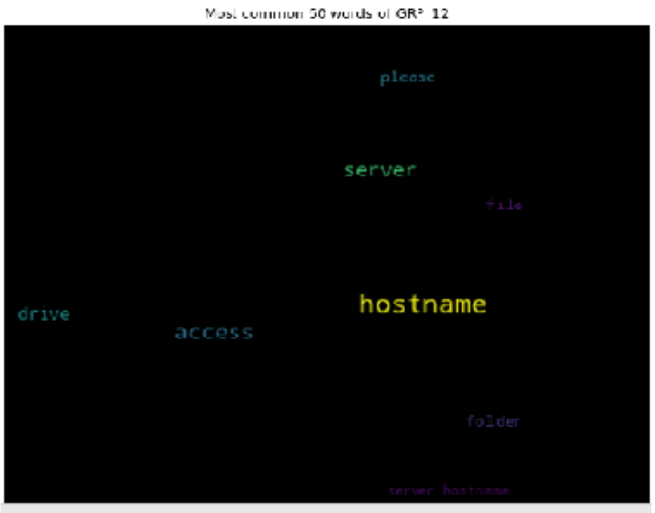
Below, we have visualized the word cloud for entire dataset and also separately for each of the assignment groups which further helped us identify the keywords/patterns which we could use for assigning the tickets to relevant groups.

Most common words for entire dataset are shown below through word cloud & graph

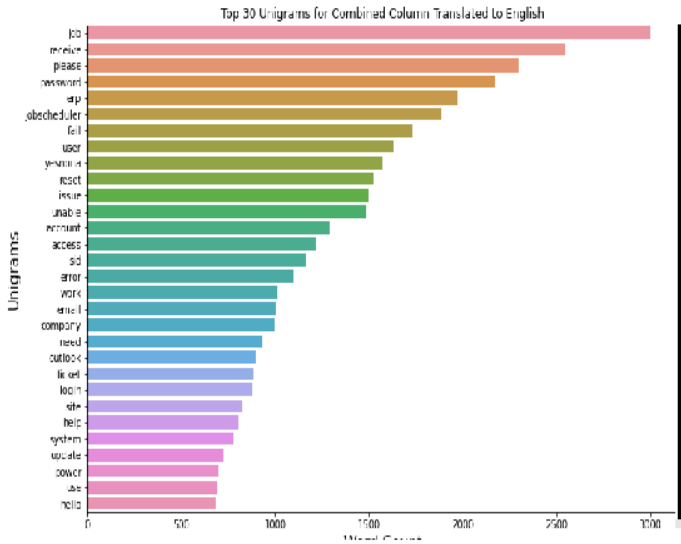


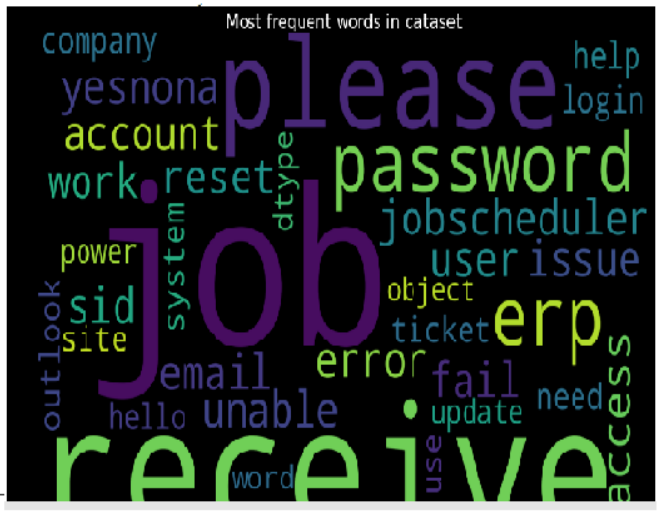
Most common 50 words for Top 6 group are plotted below (basis bigrams within each group)



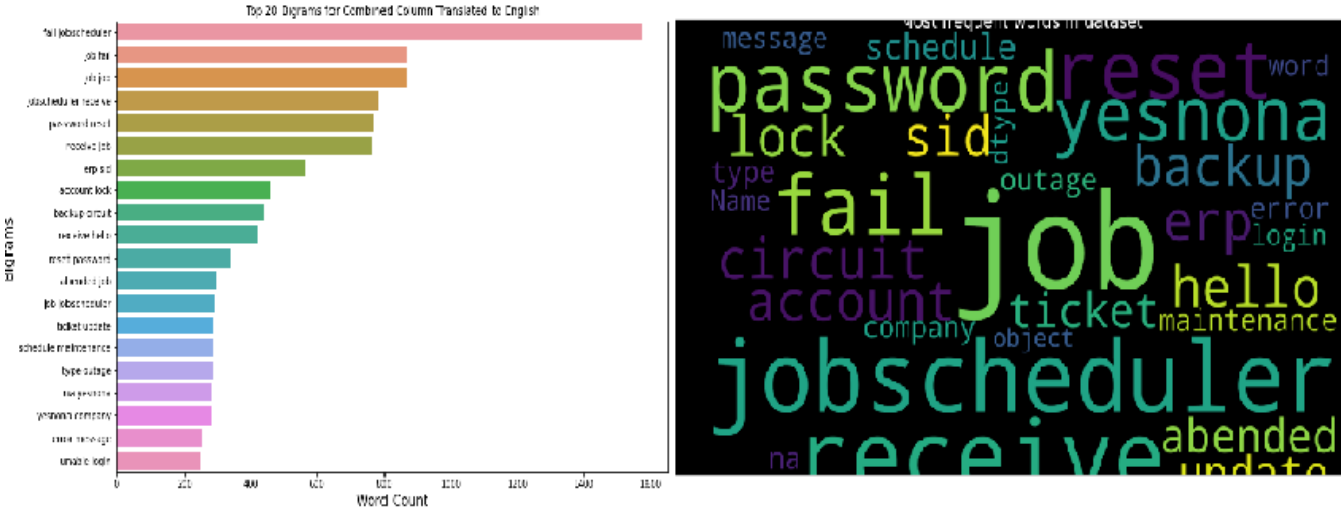


Unigram with word cloud

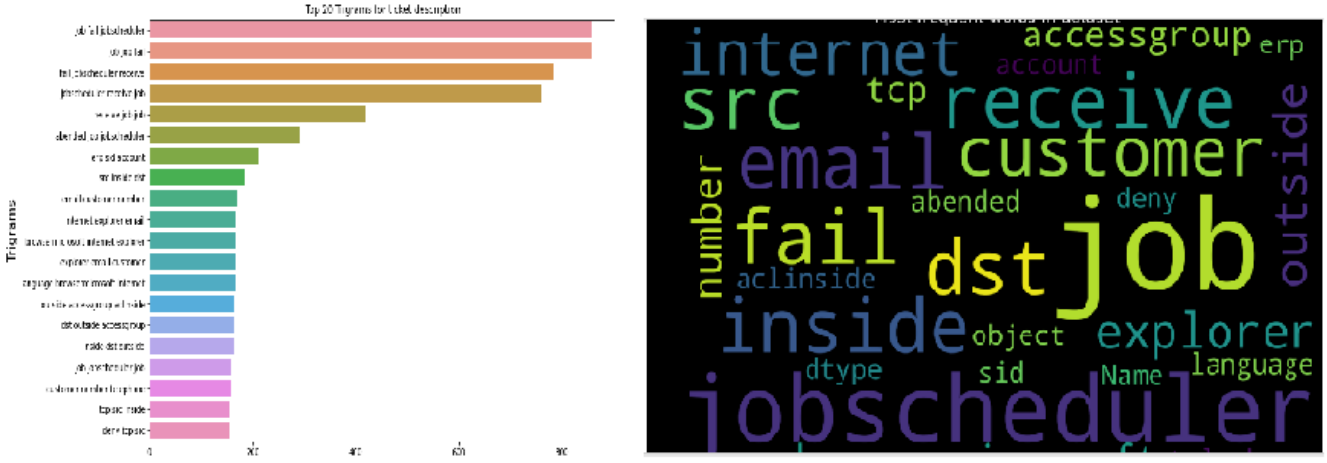




Bigram with word cloud



Trigram with word cloud



With the above visualization, it has identified the some groups can be clearly separated by defined keyword, hence we can apply deterministic rule first to identify and separate the data & rest can be taken ahead for model building.

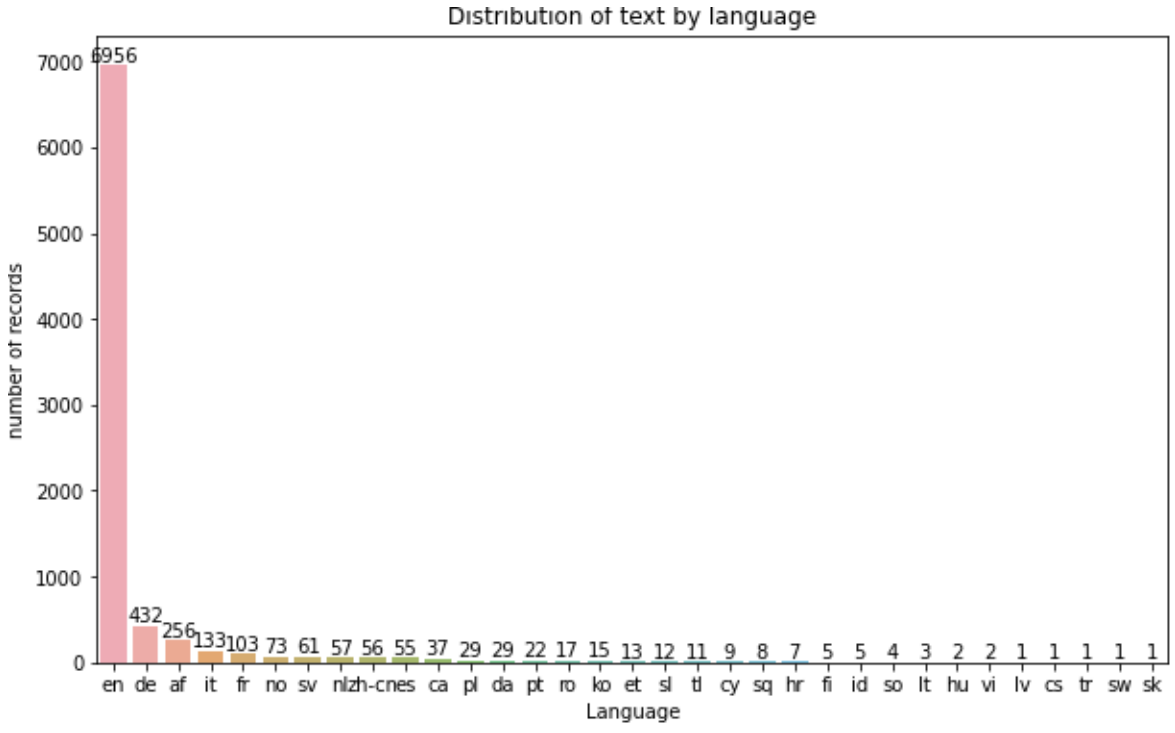
6. Understanding Language distribution

With below visualization it has identified data is distributed in different languages& hence there is need for language conversion during data pre processing. Majority of records are in Eng language.

In [27]:
```
dataset1['Language'].value_counts()
```

Out[27]:

en	6956
de	432
af	256
it	133
fr	103
no	73
sv	61
nl	57
zh-cn	56
es	55
ca	37
pl	29
da	29
pt	22



Data Pre-processing & Feature Engineering

- ✓ We realized that data contains digits, special characters, URLs, new lines, spaces etc. and hence looked for different libraries to deal with these data findings
- ✓ Explored libraries like re, stop words from NLTK, lemmatization from NLTK, contractions by pycontractions, tokenizer from NLTK, stemming by NLTK, Beautiful Soup from bs4
- ✓ Explored Google translator &go slate libraries for language translation
- ✓ Sklearn utils resample for data up sampling for minority groups
- ✓ We visualized the word clouds before and after data pre-processing to understand the high frequency words appearing in the corpus
- ✓ As we realized the both short description and description columns had different text and hence adding context to the corpus, so we decided to combine both columns before feeding to the model
- ✓ And hence we introduced a new column for combined description
- ✓ Introduced a column to document the comparison score, which is the output of fuzzywuzzy algorithm
- ✓ Had a column to capture the word counts for columns like description, short description and combined description
- ✓ One of the observations was also that, large set of incidents were created by one Caller

1. Correcting Null & duplicate

Null & duplicate data correction has been mentioned in above EDA steps

2. Treatment of garbled texts

- ✓ Garbled text that is the result of text being decoded using an unintended character encoding. The result is a systematic replacement of symbols with completely unrelated ones, often from a different writing system. Few such garbled texts (also called Mojibakes) are ¶, ç, å, €, æ, œ, º, ‡, ¼, ¥ etc.

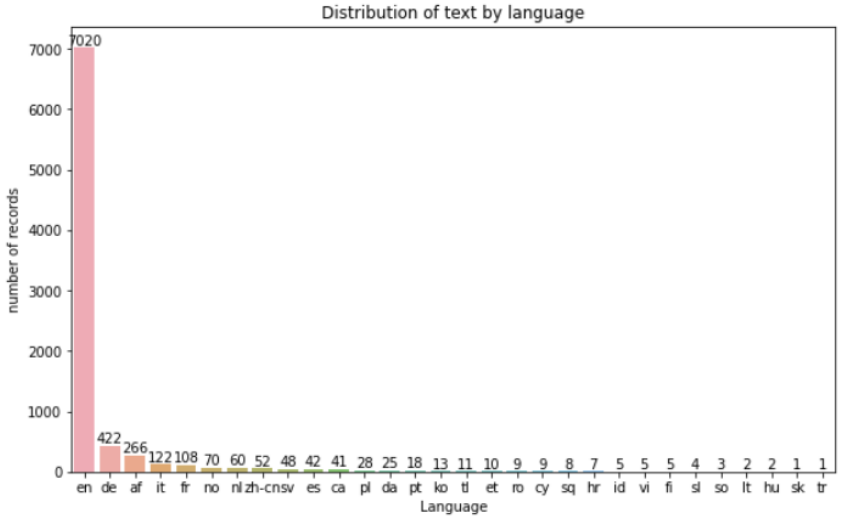
- ✓ To fix these mojibakes, we make use of ftfylibrary (version (6.0.3)) (refer appendix for details) which takes in these bad garbled unicodes and outputs the good unicodes

Example: We identified one such patterns in row#8471 and treated it as below –

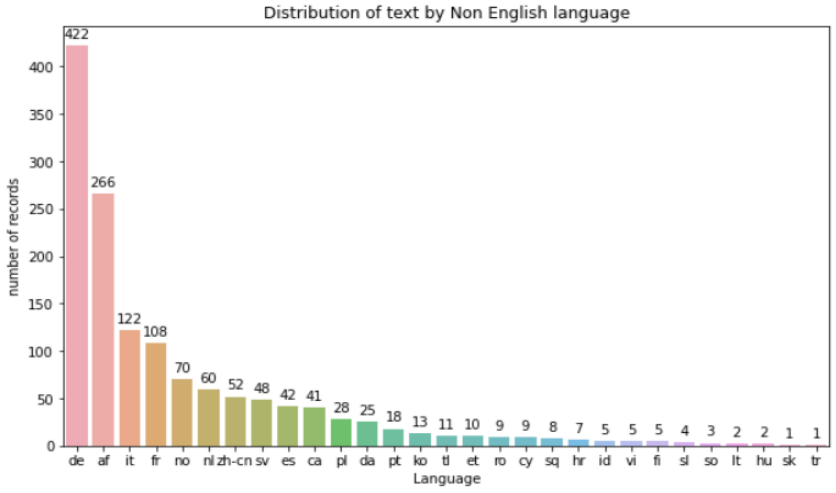
```
Garbled text: ç”µè„‘å¼€æœºå¼€ä¸å‡ºæ¥ to åºè´ï¼Œæ—©ä¸Šç”µè„‘å¼€æœºå¼€ä¸å‡ºæ¥
Fixed text: 电脑开机开不出来 to 小贺,早上电脑开机开不出来
```

Language Detection and Translation

- ✓ Detection: We use langdetect library (refer appendix for details) which helps in detecting the language of the combined description. Below is the distribution of languages we find in our data:



Below is the graph with distribution of Languages without English



Observations -

- ✓ After English, German is the 2nd most common language
- ✓ There are 10 Language which has frequency less than 10 records

Translation: As we see above, there are quite a few languages other than English, so it becomes very important to translate these other languages to English, so that our data is standardized and ready to be interpretable by the models.

To achieve this translation, below are few approaches / libraries we tried -

1. Goslate
2. Google Translate
3. Deep Translator
4. BingTranslator

All the above approaches had some issues or challenges pertaining to

i. Limitation on number of attempts to translate using an IP
ii. After a certain number of attempts, few packages provide only paid service
iii. Limitation on number of language support. For Eg - Google Goslate does not support a few languages.

To achieve this translation, we make use of **Goslate library(refer appendix for details)**which uses the Google Translate Ajax API to make calls to such methods as detect and translate.

We ran our code in batches on different machines to get the translations of all the supported languages. We marked few as "no" for those sentences which could not be translated.
To verify the translation's correctness, we also checked in Google Sheets > Translate function.
Service urls used:
translate.google.com, translate.google.com.au, translate.google.com.ar, translate.google.co.kr, translate.google.co.in, translate.google.co.jp, translate.google.at, translate.google.de, translate.google.ru, translate.google.ch, translate.google.fr, translate.google.es, translate.google.ae

Example:

```
Original Text :  an mehreren pc`s lassen sich verschiedene
prgramdntyme nicht ffnen. an mehreren pc`s lassen sich
verschiedene prgramdntyme nicht ffnen. bereich cnc.
Traslated to English :  Several prgramdntyme can not be folded
on several PCs. Several prgramdntyme can not be folded on
several PCs. Area CNC.
```

3. Text treatment

For further standardization and easier pattern recognition, We apply functions to treat the text for following:

- ✓ Upper to lower case conversion
- ✓ Removing the numbers
- ✓ Replacing punctuations with blank space
- ✓ Replace multiple spaces to single
- ✓ Stop words removal (only for ML models, not for Deep Learning models)

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
stopwords = set(stopwords.words('english'))
 # Remove stopwords
df_ML['combined_description'] = df_ML['combined_description'].apply(lambda x: ' '.join([word for word in x.split() if word not in stopwords]))
```

```
Ticket_desc = df_ML['combined_description']
#Define empty list
ticket_desc_cleaned = []
res = []
#Define for loop to iterate through the elements of the ticket_desc
for l in Ticket_desc:
    #Parse the contents of the cell
    soup = BeautifulSoup(l, 'html.parser')
    #Find all instances of the text within the </p> tag
    for el in soup.find_all('p'):
        res.append(el.get_text())
    #concatenate the strings from the list
    endstring = ' '.join(map(str, res))
    #reset list
    res = []
    #Append the concatenated string to the main list
    ticket_desc_cleaned.append(endstring)
```

```
def fn_remove_irrelaventWords_LevelTwo(df,columnName):
  for index in range(df.shape[0]):
    df[columnName][index] = df[columnName][index].lower()
    df[columnName][index] = df[columnName][index].replace("// ::",' ')
    df[columnName][index] = df[columnName][index].replace("<",' ')
    df[columnName][index] = df[columnName][index].replace(">",' ')
    df[columnName][index] = df[columnName][index].replace(";",' ')
    df[columnName][index] = df[columnName][index].replace("-",' ')
    df[columnName][index] = df[columnName][index].replace("•",' ')
    df[columnName][index] = df[columnName][index].replace("?",' ')
    df[columnName][index] = df[columnName][index].replace("\\",' ')
    df[columnName][index] = df[columnName][index].replace("\/",' ')
    df[columnName][index] = df[columnName][index].replace(":",' ',)
    df[columnName][index] = df[columnName][index].replace("%",' ',)
    df[columnName][index] = df[columnName][index].replace("-",' ',)
    df[columnName][index] = df[columnName][index].replace("[mail ]",' ')
    df[columnName][index] = df[columnName][index].replace("[",' ')
    df[columnName][index] = df[columnName][index].replace("]",' ')
    df[columnName][index] = df[columnName][index].replace("< mail >",' ')
    df[columnName][index] = df[columnName][index].replace("+",' ')
    df[columnName][index] = df[columnName][index].replace("\"",' ')
    df[columnName][index] = df[columnName][index].replace("' ",' ')
    df[columnName][index] = df[columnName][index].replace(" '",' ')
    df[columnName][index] = df[columnName][index].replace("* * * ",' ')
    df[columnName][index] = df[columnName][index].replace(" * * *",' ')
```

```
df_lang_clean = fn_remove_irrelaventWords_LevelTwo(df_lang,"combined_description")
```

```
df_lang_clean.head(20)
```

Below is the example of row#50 in our data, with original text and treated text

```
Original text:
job mm_zscr0099_dly_merktc3 failed in job_scheduler at: 10/31/2016 08:36:00 received from: monitoring_tool@company.com

job mm_zscr0099_dly_merktc3 failed in job_scheduler at: 10/31/2016 08:36:00
_____________________________________________________________________________________________
Cleaned text:
job mmzscrdlymerktc failed in jobscheduler at received from monitoring_tool@company.com job mmzscrdlymerktc failed in jobscheduler at
```

4. Stemming and Lemmatization (only for ML models, not required for Deep Learning models)

Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language.

Lemmatization, unlike Stemming, reduces the inflected words properly ensuring that the root word belongs to the language. In Lemmatization root word is called Lemma. A lemma (plural lemmas or lemmata) is the canonical form, dictionary form, or citation form of a set of words.

We make use of spacy library (version 2.2.4) (refer appendix for details)to help with lemmatization of our text data.

Example: Below is an example of row#50 in our dataset with original and lemmatized text

```
Original text:
job mmzscrdlymerktc failed jobscheduler received monitoring_tool@company.com job mmzscrdlymerktc failed jobscheduler
_____________________________________________________________________________________________
Lemmatized text:
job mmzscrdlymerktc fail jobscheduler receive monitoring_tool@company.com job mmzscrdlymerktc fail jobscheduler
```

Deterministic Rule & Handling Data imbalance

Simple deterministic rules can be applied from processes that are already in place. We have identified a few keywords which depict incidents/tickets belonging to that specific group. Below are a few examples belonging to specific groups. As we see combination of Caller keyword & description “network outage” or “circuit outage” are most frequent occurring in GRP 8

Some data was identified identical & hence grouped

Basis below rule known data got separated & rest was feed to the model

description keyword	Desc Rule	Caller Rule	caller keyword	Group
telephony software	contain	contain	-	GRP_7
cutview	contain	contain	-	GRP_66
engg application	contain	contain	-	GRP_58
ethics	contain	contain	-	GRP_23
crm dynamics	contain	contain	-	GRP_22
distributor tool	contain	contain		GRP_21
company center	contain	contain		GRP_21
network outage or circuit outage	contain	contain	bpctwhsn kzqsbmtp	GRP_8

Rules defined along with count of ticket in the group which are most frequent

```
def deterministicRules(df,columnName):
  for i in range(df.shape[0]):
      #1 Contains telephony_software > GRP_7
    if pd.notna(df[columnName][i]):
      if ('telephonysoftware' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_7'
      #2 contains cutview >  GRP_66
      elif ('cutview' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_66'
      #3 contains engg application >  GRP_58
      elif ('engg application' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_58'
      #4 contains ethics >  GRP_23
      elif ('ethics' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_23'
      # contains crm dynamics >  GRP_22
      elif ('crm dynamics' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_22'
      # contains distributor tool & company center >  GRP_21
      elif ('distributor tool' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_21'
      elif ('company center' in df[columnName][i]):
        df['pred_group'][i] = 'GRP_21'
      # contains bpctwhsn kzqsbmtp & network outage or circuit outage >  GRP_8
      elif (df['Caller'][i] == 'bpctwhsn kzqsbmtp' and ('network outage' in df[columnName][i] c
```

```
df_determinted_ML['pred_group'].value_counts()
```

```
GRP_0     125
GRP_7      83
GRP_2      27
GRP_21     21
GRP_22     15
```

```
df_NonDet.insert(loc=4,column='New Assignment Group',value=np.nan,allow_duplicates=True)
```

```
groupsToBeMerged = pd.DataFrame(df_NonDet['Assignment group'].value_counts() <=10)
groupsToBeMerged = groupsToBeMerged[groupsToBeMerged['Assignment group'] == True]
groupsToBeMergedList = list(groupsToBeMerged.index)
groupsToBeMergedList
```

```
['GRP_52',
 'GRP_51',
 'GRP_55',
 'GRP_65',
 'GRP_59',
 'GRP_49',
 'GRP_46',
```

Treating Class Imbalance

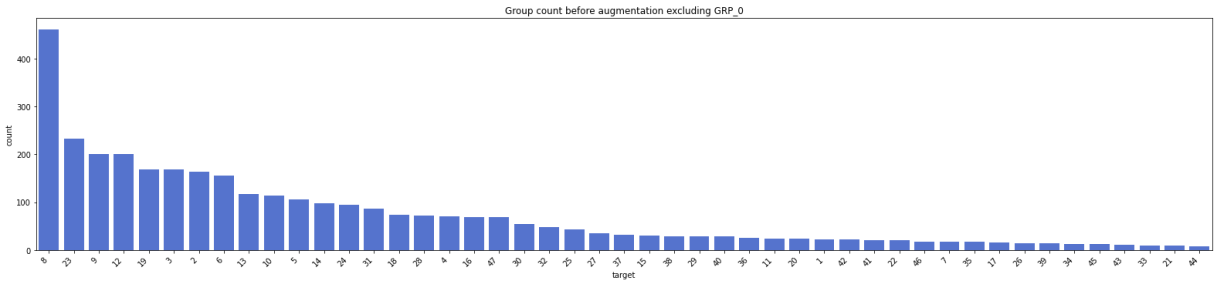
We tried proceeding with modelling without augmentation, but were getting low accuracy with some Over fitting as well. So we tried treating the class imbalance through two methods:

1. Data Augmentation techniques
2. Data up sampling/resampling techniques

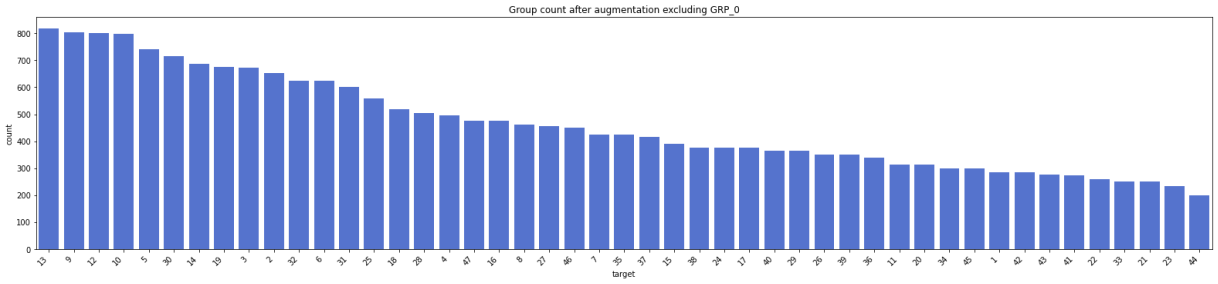
Data Augmentation:

- ✓ Data augmentation for treating Class imbalance to improve model performance
- ✓ As we highlighted in EDA above, there is severe class imbalance which is likely to affect the model performance. So it's very important to treat this
- ✓ There are many methods to solve for this - resampling and augmentation being some of those
- ✓ Augmentation can be done either on character, word or entire sentences. Here we opted for word augmentation.
- ✓ We can make use of word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), fasttext (Joulin et al., 2016), BERT(Devlin et al., 2018) and wordnet to insert and substitute similar word. Word2vecAug, GloVeAug and FasttextAug use word embeddings to find most similar group of words to replace original word. On the other hand, BertAug use language models to predict possible target word.
- ✓ Here, we have proceeded with **WordNetAug**, which use statistics way to find the similar group of words/synonyms. The library that we use is **nlpaug (version 1.1.8)(refer appendix for details)**
- ✓ Through this augmentation method, we replace three words with their synonyms in a record, and create multiple synonymous versions of the records
- ✓ Below is the distribution of classes (excluding Group 0) before and after augmentation

Distribution of classes before augmentation (excluding group 0):



Distribution of classes after augmentation (excluding group 0):



Example of augmentation (to assess quality of sentences)

Original text:

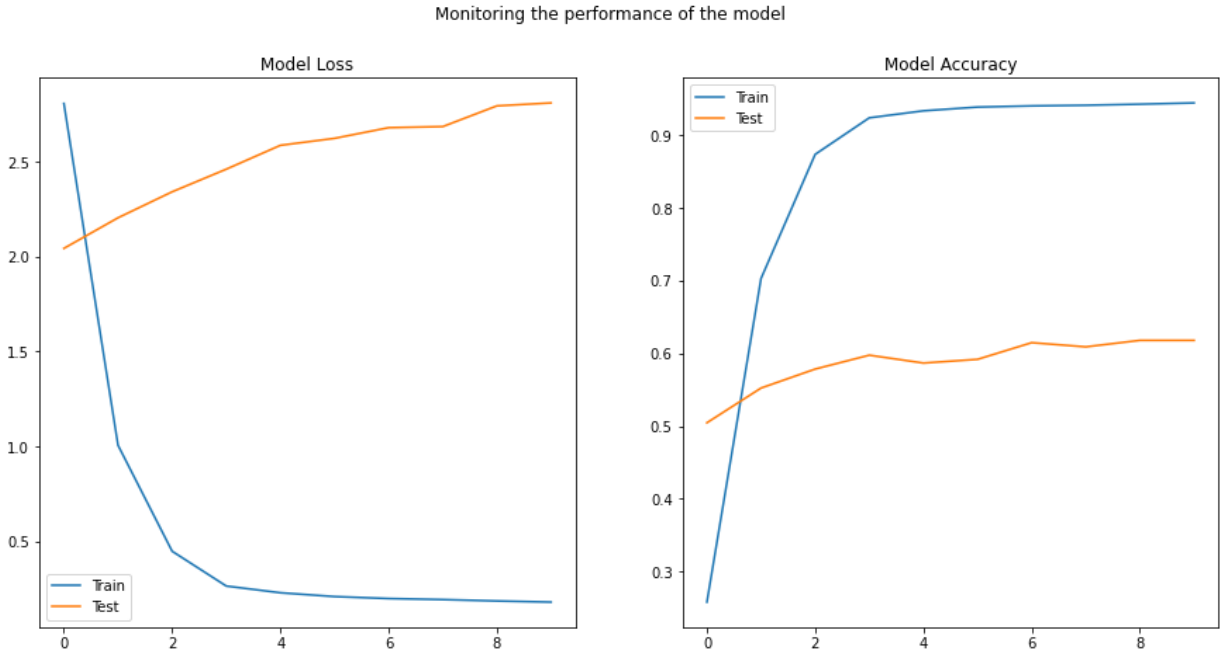
unable to log in to engineering tool and skype unable to log in to engineering tool and skype

Augmented text:

unable to log in to engineering tool and skype ineffectual to lumber in to engineering shaft and skype unable to log in to engineering tool and skype unable to log in to applied science tool and skype

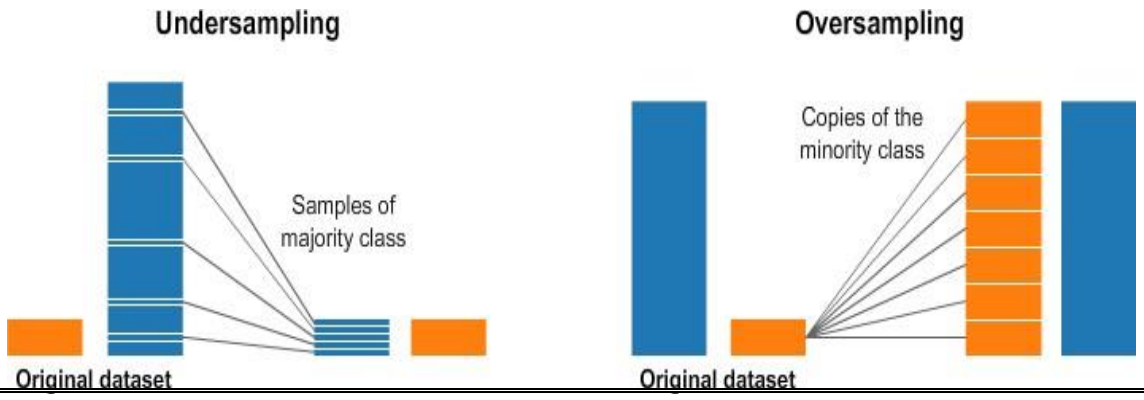
Impact of augmentation:

We saw an increase in training dataset accuracy but the test accuracy was still low indicating a high overfitting.



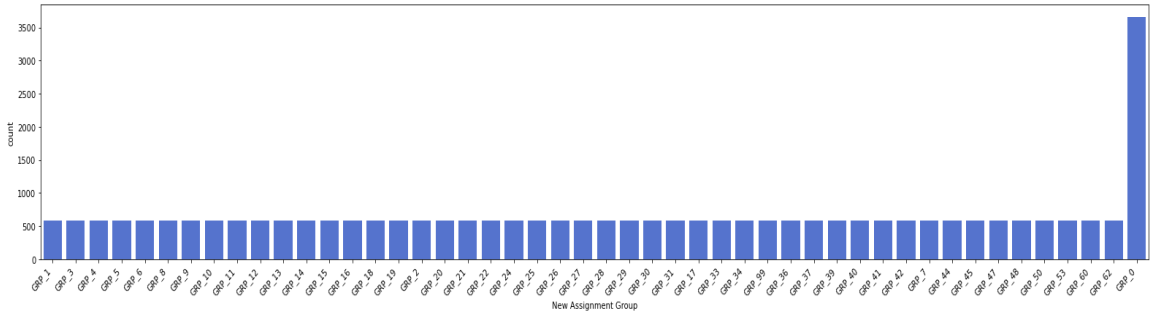
b. Data Upsampling/Resampling:

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and / or adding more examples from the minority class (over-sampling).



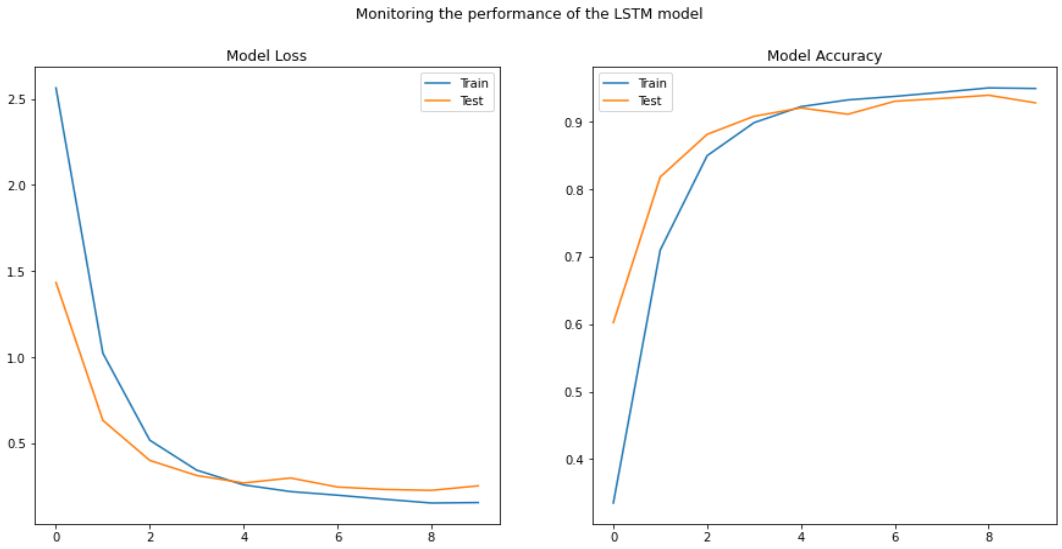
Steps:

- ✓ Here, we tried resampling through replacement on all groups except group 0 (as group 0 has sufficient observations)
- ✓ We upsampled the counts of all groups to match that of the second largest group, since the largest group (group 0) has extremely high number of records (so we didn't match with the largest group)
- ✓ We used **resample** package from **sklearn.utils library** **(refer appendix for details)**
- ✓ Below is how the class distribution looks like post Resampling (the last group is group 0, excluding that all other upsampled to same sample size as that of second largest group)



Impact of upsampling via resampling:

We saw an increase in both training dataset accuracy and test dataset accuracy. There was no evidence of any overfitting. So we proceeded with treating class imbalance through resampling



Derive Train, test and Validation data split

- ✓ We went 80:20 ratios for splitting the data set into train & test data sets
- ✓ Used validation split of ration 0.2(20%) during model training (model.fit)
- ✓ Used standard test_train split library for splitting the data

```
from sklearn.model_selection import train_test_split

# Create training and test datasets with 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(df_ML_NonAug.combined_description  ,
                                                    df_ML_NonAug.target,
                                                    test_size=0.20,
                                                    random_state=42)
print('\033[1mShape of the training set:\033[0m', X_train.shape, y_train.shape)
print('\033[1mShape of the test set:\033[0m', X_test.shape, y_test.shape)
```

Identify the relevant Algorithms

Firstly, finalized that it's a text-based problem and hence decided to categorize under the NLP methodology.

We have worked with different algorithms

- ✓ Identified the given problem is a multi-class classification problem with presence of 74 classes in the data set
- ✓ Listed down various classifiers among supervised, unsupervised and neural networks which work better for text-based classification problem

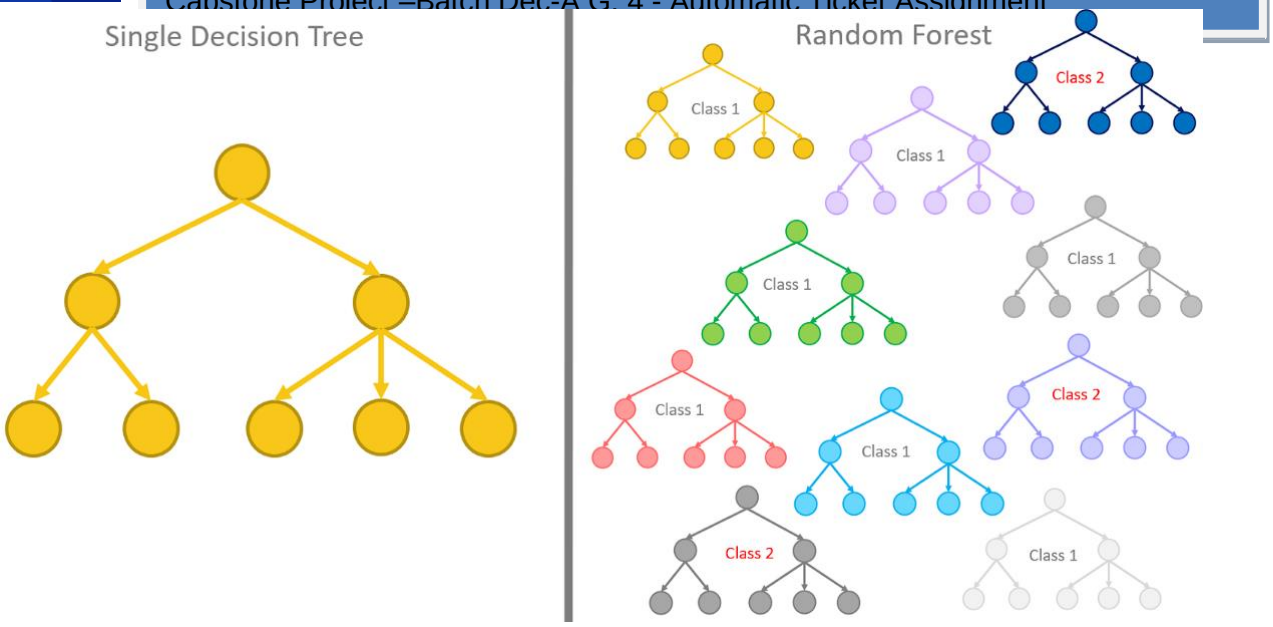
Model Building & Model tuning

1. Random Forests
2. XGBoost
3. SVC
4. k-NN
5. Naïve Bayes were selected.

All models help in classification problems. Hence, the selection

Random Forests

Random forests or random decision forests are an ensemble learning method for classification, also used in regression. Algorithm creates multiple of decision trees at training time and output of the class is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



Advantages:

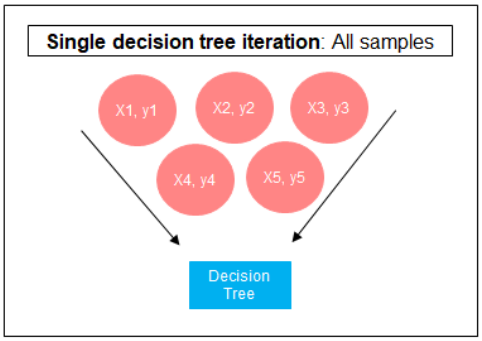
1. It reduces overfitting in decision trees and helps to improve the accuracy
2. It is flexible to both classification and regression problems
3. It works well with both categorical and continuous values
4. It automates missing values present in the data
5. Normalizing of data is not required as it uses a rule-based approach.'

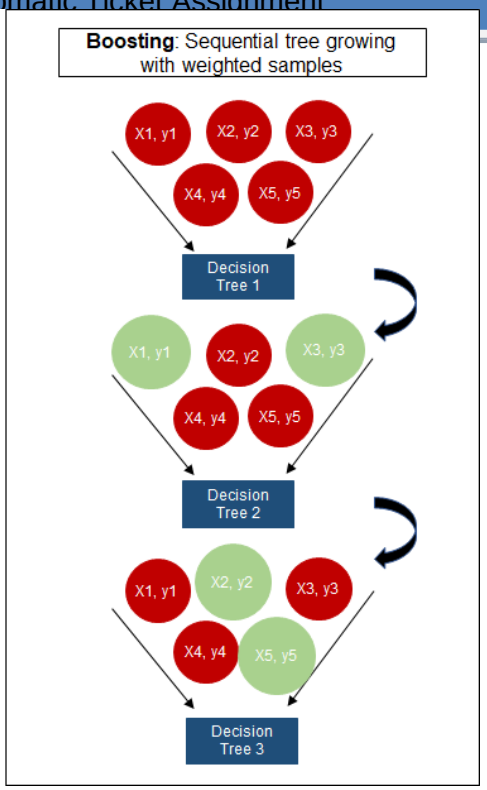
Disadvantages:

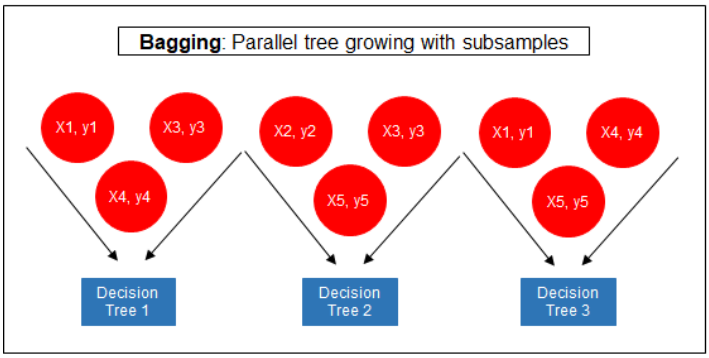
1. It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
2. It also requires much time for training as it combines a lot of decision trees to determine the class.
3. Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

XGBoost

XGBoost stands for eXtreme Gradient Boosting.
Boosting is an ensemble learning technique to build a strong classifier from several weak classifiers in series. Boosting algorithms play a crucial role in dealing with bias-variance trade-off.`
XGBoost is an ensemble learning algorithm meaning that it combines the results of many models, called base learners to make a prediction. Like Random Forests, XGBoost uses Decision Trees as base learners.
Individual decision trees are low-bias, high-variance models. They are incredibly good at finding the relationships in any type of training data but struggle to generalize well on unseen data.







XGBoost Features

Regularized Learning: Regularization term helps to smooth the final learnt weights to avoid over-fitting. The regularized objective will tend to select a model employing simple and predictive functions.

Gradient Tree Boosting: The tree ensemble model cannot be optimized using traditional optimization methods in Euclidean space. Instead, the model is trained in an additive manner.

Shrinkage and Column Subsampling: Besides the regularized objective, two additional techniques are used to further prevent overfitting.

- ✓ The first technique is shrinkage introduced by Friedman. Shrinkage scales newly added weights by a factor η after each step of tree boosting. Shrinkage reduces the influence of each tree and leaves space for future trees to improve the model.
- ✓ The second technique is the column (feature) subsampling. This technique is used in Random Forest. Column sub-sampling prevents over-fitting even more so than the traditional row sub-sampling. The usage of column sub-samples also speeds up computations of the parallel algorithm.

Advantages:

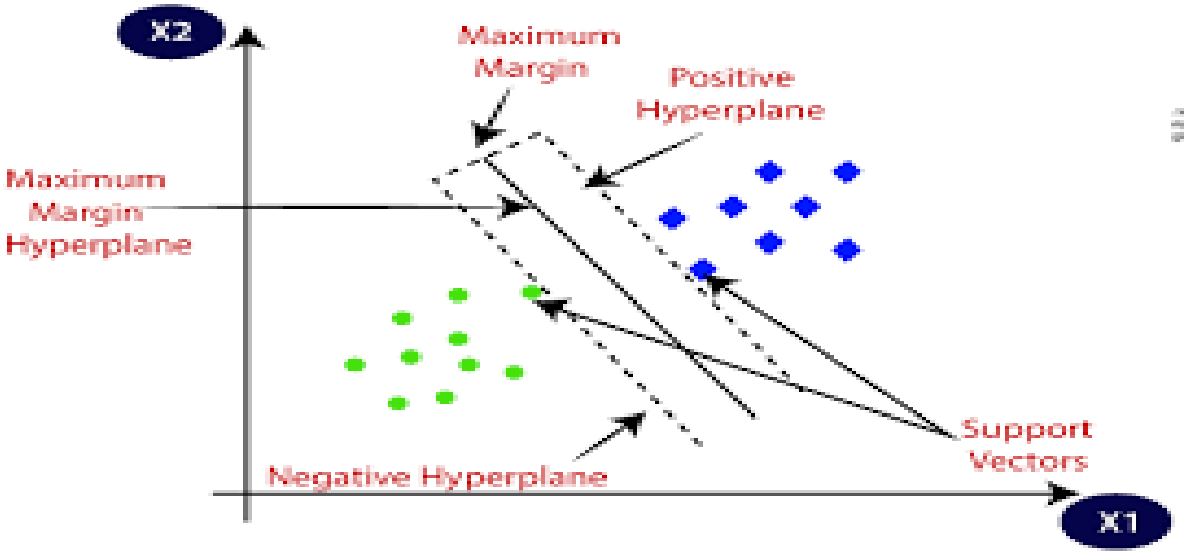
1. Less feature engineering required (No need for scaling, normalizing data, can also handle missing values well)
2. Feature importance can be found out(it output importance of each feature, can be used for feature selection)
3. Fast to interpret
4. Outliers have minimal impact.
5. Handles large sized datasets well.
6. Good Execution speed
7. Good model performance (wins most of the Kaggle competitions)
8. Less prone to overfitting

Disadvantages:

1. Difficult interpretation, visualization tough
2. Overfitting possible if parameters not tuned properly.
3. Harder to tune as there are too many hyper parameters.

Linear SVC

The Support Vector Machine (SVM) technique is a popular and highly accurate machine learning method for classification problems. SVM try to find an optimal hyperplane within the input space to correctly classify the binary (or multi-class classification problem.



Advantages:

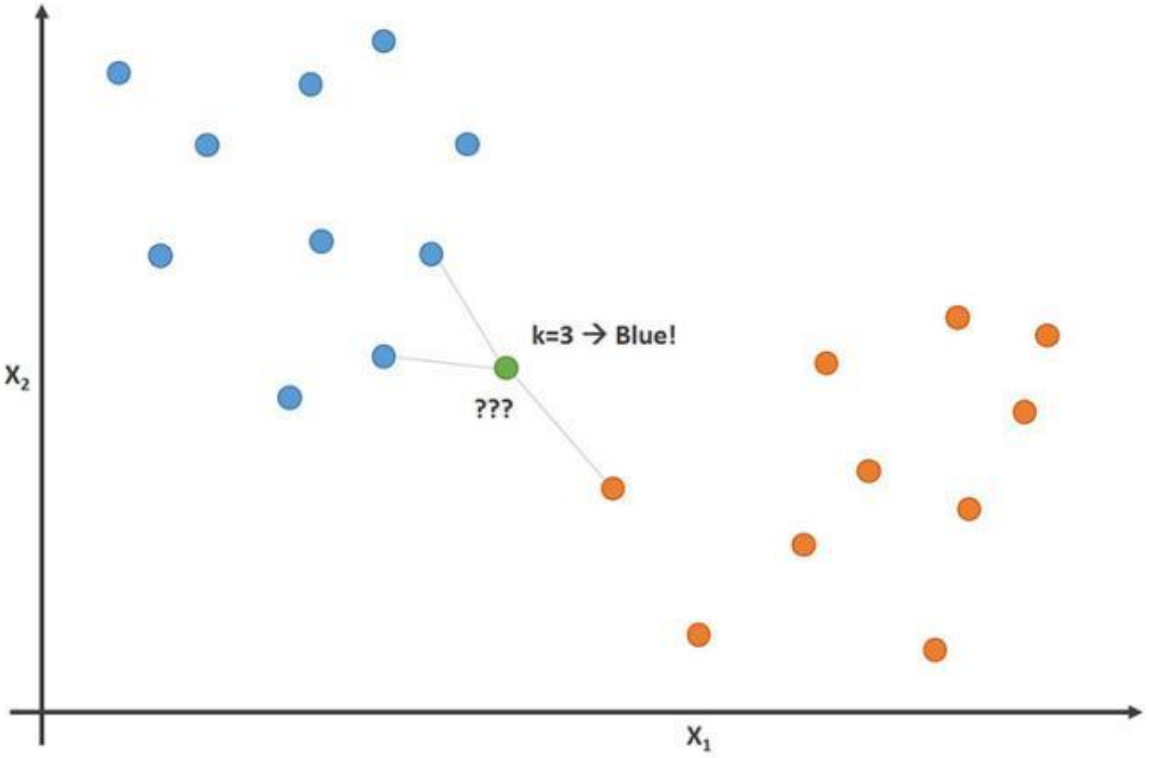
1. SVM works relatively well when there is a clear margin of separation between classes.
2. SVM is more effective in high dimensional spaces.
3. SVM is effective in cases where the number of dimensions is greater than the number of samples.
4. SVM is relatively memory efficient
5. Outliers have less impact.

Disadvantages:

1. SVM algorithm is not suitable for large data sets.
2. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
3. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
4. As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

KNN

K Nearest Neighbors is a Supervised Learning algorithm and is mostly used for classification and also regression. KNN algorithm is uses distances such as Euclidean, Manhattan and Minkowski to find the K nearest neighbors in the training data and then uses these labels to predict



Advantages:

1. Simple to understand and implement
2. No assumption about data (for e.g. in case of linear regression we assume dependent variable and independent variables are linearly related, in Naïve Bayes we assume features are independent of each other etc., but k-NN makes no assumptions about data)
3. Constantly evolving model: When it is exposed to new data, it changes to accommodate the new data points.
4. Lazy algorithms do not need any training data points for model generation
5. Multi-class problems can also be solved.
6. One Hyper Parameter: K-NN might take some time while selecting the first hyper parameter but after that rest of the parameters are aligned to it.
7. All training data used in the testing phase. Hence. training faster and testing phase slower and costlier leading to more memory usage and also more time.

Disadvantages:

1. Slow for large datasets.
2. Curse of dimensionality: Does not work very well on datasets with large number of features.
3. Scaling of data absolute must.
4. Does not work well on Imbalanced data. So before using k-NN either under sample majority class or oversample minority class and have a balanced dataset.
5. Sensitive to outliers.
6. Can't deal well with missing values

Naïve Bayes

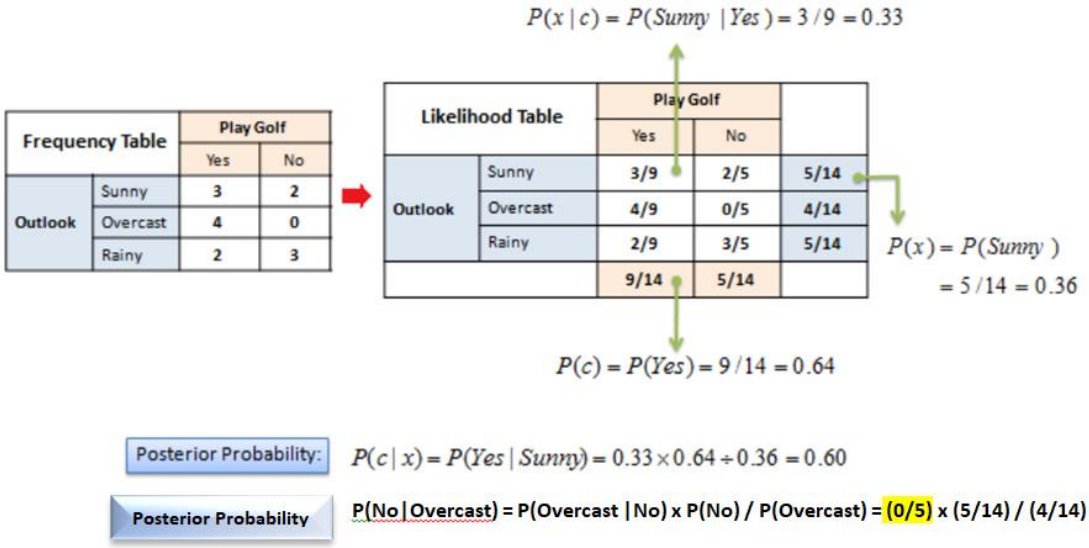
A Naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points. The key insight of Bayes' theorem is that the probability of an event can be adjusted as new data is introduced. These classifiers are widely used for machine learning because they are simple to implement.

Advantages:

1. Real time predictions: It is very fast and can be used in real time.
2. Scalable with Large datasets
3. Insensitive to irrelevant features.
4. Multi class prediction is effectively done in Naive Bayes
5. Good performance with high dimensional data(no. of features is large)

Disadvantages:

1. Independence of features does not hold:
The fundamental Naive Bayes assumption is that each feature makes an independent and equal contribution to the outcome. However this condition is not met most of the times.
2. Bad estimator:
Probability outputs from predict_proba are not to be taken too seriously.
3. Training data should represent population well:
If you have no occurrences of a class label and a certain attribute value together (e.g. class="No", shape="Overcast ") then the posterior probability will be zero. So if the training data is not representative of the population, Naive bayes does not work well.(This problem is removed by smoothening techniques).



Machine Learning Model Results.

Model	accuracy_training	accuracy_test	recallscore_training	recallscore_test	precision_training	precision_test	f1score_training	f1score_test	Elapsed
Random Forest	95.99	94.78	95.99	94.78	97.56	96.12	96.29	94.96	27.19
Xgboost	89.61	86.82	89.61	86.82	91.58	88.52	89.80	86.78	172.62
SVC	94.92	92.33	94.92	92.33	96.41	93.46	95.21	92.41	2.79
KNN	91.17	87.40	91.17	87.40	92.91	88.68	91.37	87.21	19.85
Naive Bayes	72.65	69.26	72.65	69.26	86.99	85.55	74.34	71.05	1.16

Deep learning models

1.LSTM

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies.
LSTMs are explicitly designed to avoid the long-term dependency problem.
The popularity of LSTM is due to the gating mechanism involved with each LSTM cell.
In a normal RNN cell, the input at the time stamp and hidden state from the previous time step is passed through the activation layer to obtain a new state.
Whereas in LSTM the process is slightly complex, as you can see in the architecture. At each time it takes input from three different states like the current input state, the short term memory from the previous cell and lastly the long term memory.

These cells use the gates to regulate the information to be kept or discarded at loop operation before passing on the long term and short term information to the next cell. There are a total of three gates that LSTM uses as Input

Gate, Forget Gate, and Output Gate.

Input Gate

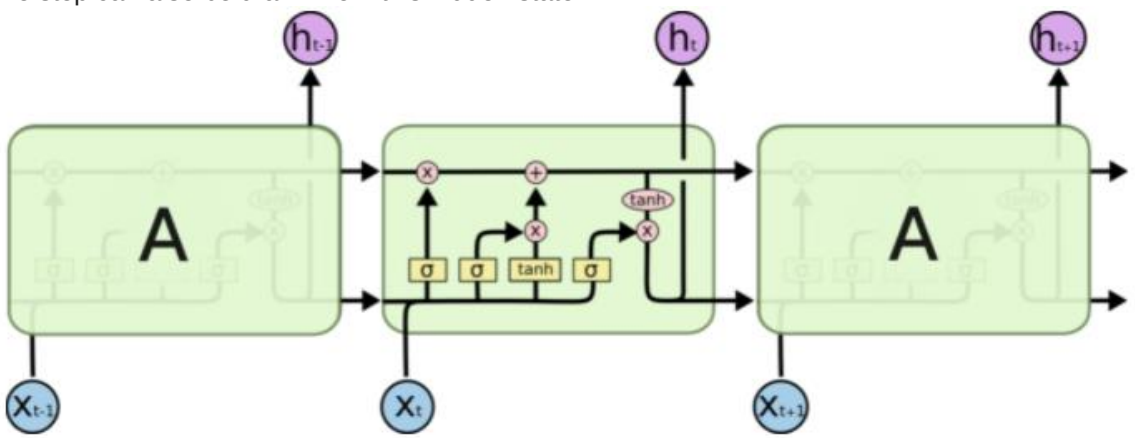
The input gate decides what information will be stored in long term memory. It only works with the information from the current input and short term memory from the previous step. At this gate, it filters out the information from variables that are not useful.

Forget Gate

The forget decides which information from long term memory be kept or discarded and this is done by multiplying the incoming long term memory by a forget vector generated by the current input and incoming short memory.

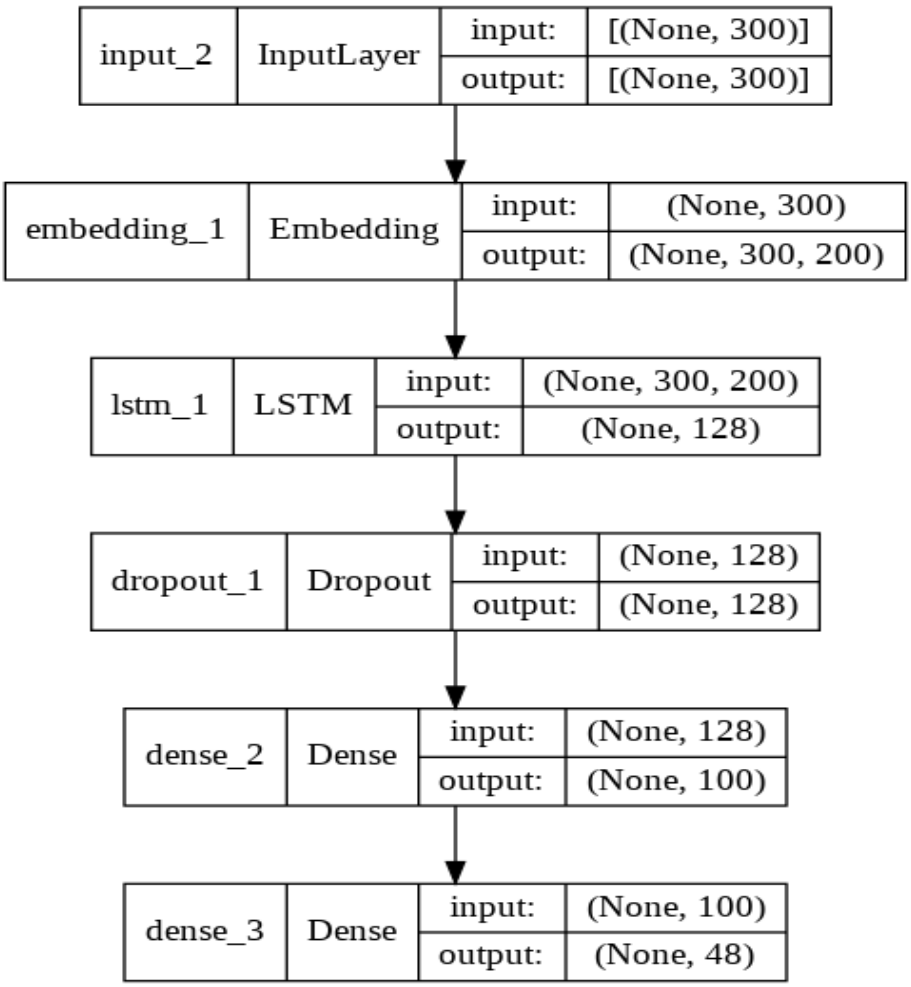
Output Gate

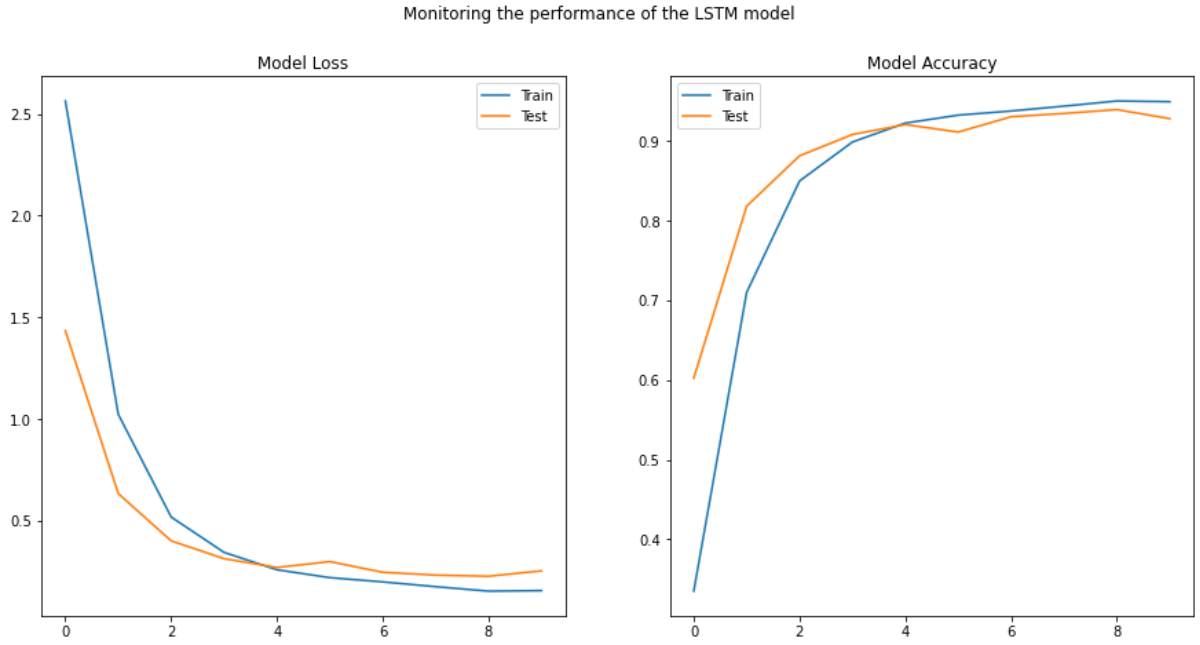
The output gate will take the current input, the previous short term memory and newly computed long term memory to produce new short term memory which will be passed on to the cell in the next time step. The output of the current time step can also be drawn from this hidden state.



Model Result LSTM

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_2 (InputLayer)         [(None, 300)]             0

 embedding_1 (Embedding)      (None, 300, 200)          2340600

 lstm_1 (LSTM)                (None, 128)               168448

 dropout_1 (Dropout)          (None, 128)               0

 dense_2 (Dense)              (None, 100)               12900

 dense_3 (Dense)              (None, 48)                4848

=================================================================
Total params: 2,526,796
Trainable params: 2,526,796
Non-trainable params: 0
```


input_2
InputLayer
input:
output:
[(None, 300)]
[(None, 300)]
embedding_1
Embedding
input:
output:
(None, 300)
(None, 300, 200)
lstm_1
LSTM
input:
output:
(None, 300, 200)
(None, 128)
dropout_1
Dropout
input:
output:
(None, 128)
(None, 128)
dense_2
Dense
input:
output:
(None, 128)
(None, 100)
dense_3
Dense
input:
output:
(None, 100)
(None, 48)

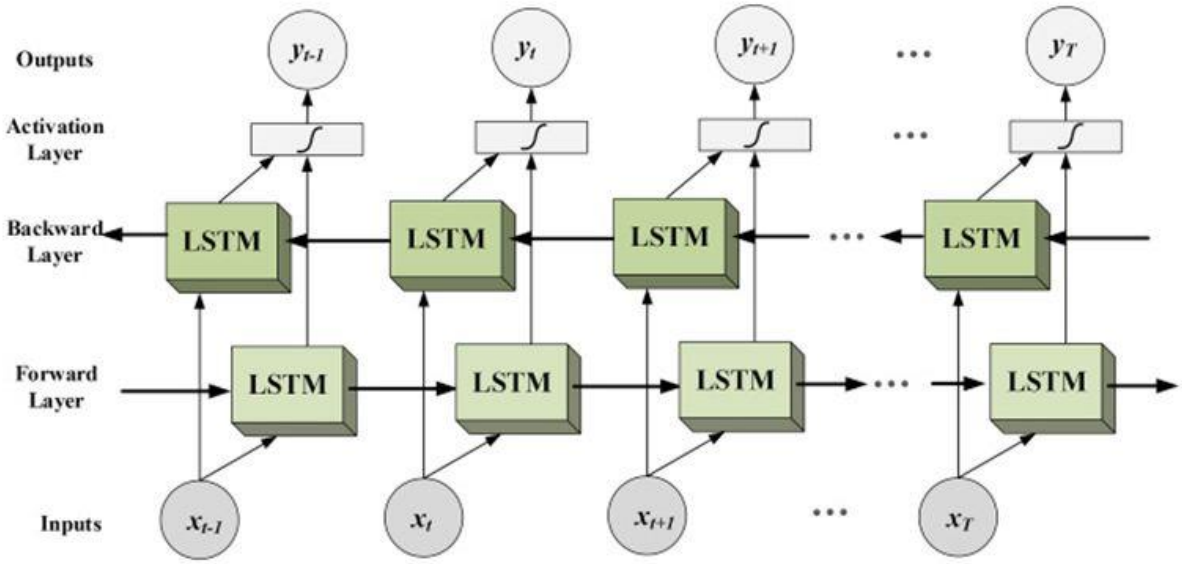


2.Bi-LSTM:(Bi-directional long short term memory):

Bidirectional recurrent neural networks(RNN) are really just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at every time step.

Using bidirectional will run the inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backward you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

In NLP sometimes to understand a word we need not just to the previous word, but also the next word.



Advantages:

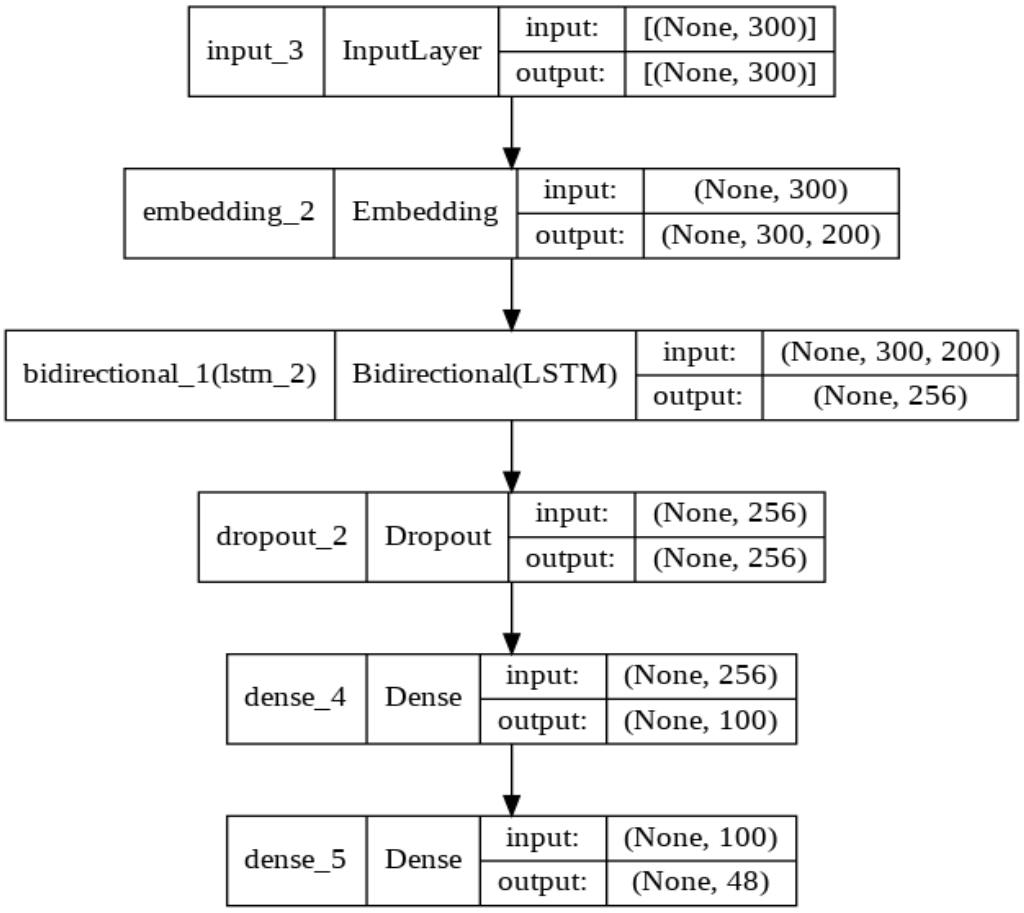
It solves the problem of fixed sequence to sequence prediction. Vanilla RNN has a limitation where both input and output has the same size.

Disadvantages:

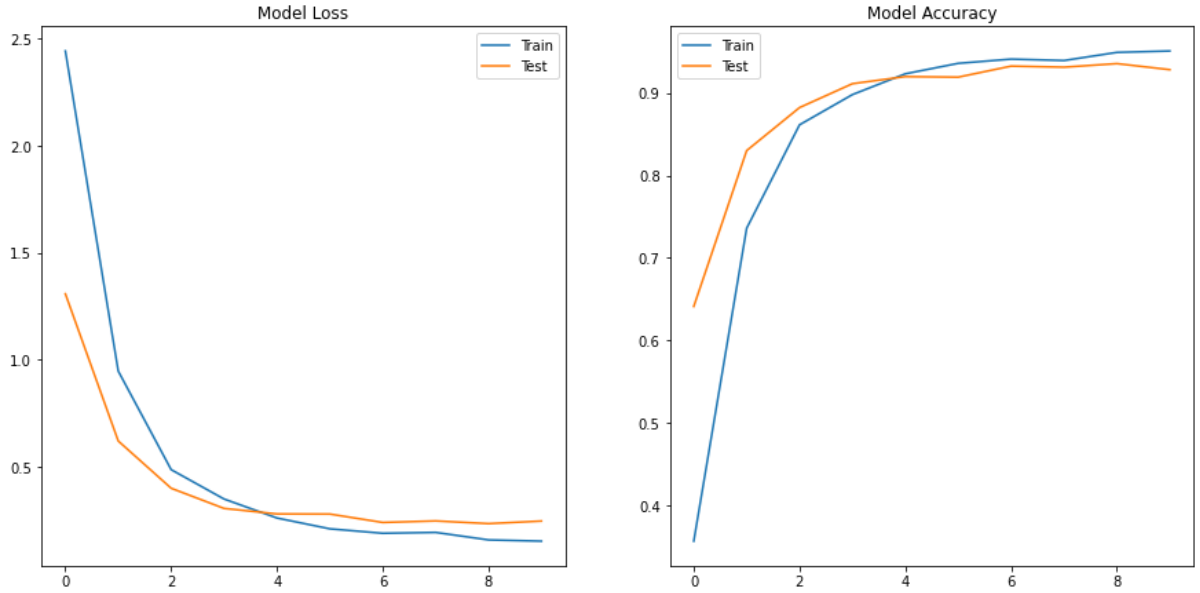
1. Since BiLSTM has double LSTM cells so it is costly.
2. Not Good fit for Speech Recognition

Model Result Bi-LSTM

```
Model: "model_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, 300)]             0

 embedding_2 (Embedding)     (None, 300, 200)          2340600

 bidirectional_1 (Bidirectio  (None, 256)              336896
 nal)

 dropout_2 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 100)               25700

 dense_5 (Dense)             (None, 48)                4848

=================================================================
Total params: 2,708,044
Trainable params: 2,708,044
Non-trainable params: 0
```



Monitoring the performance of the Bi-LSTM model



3.Gated Recurrent Unit (GRU)

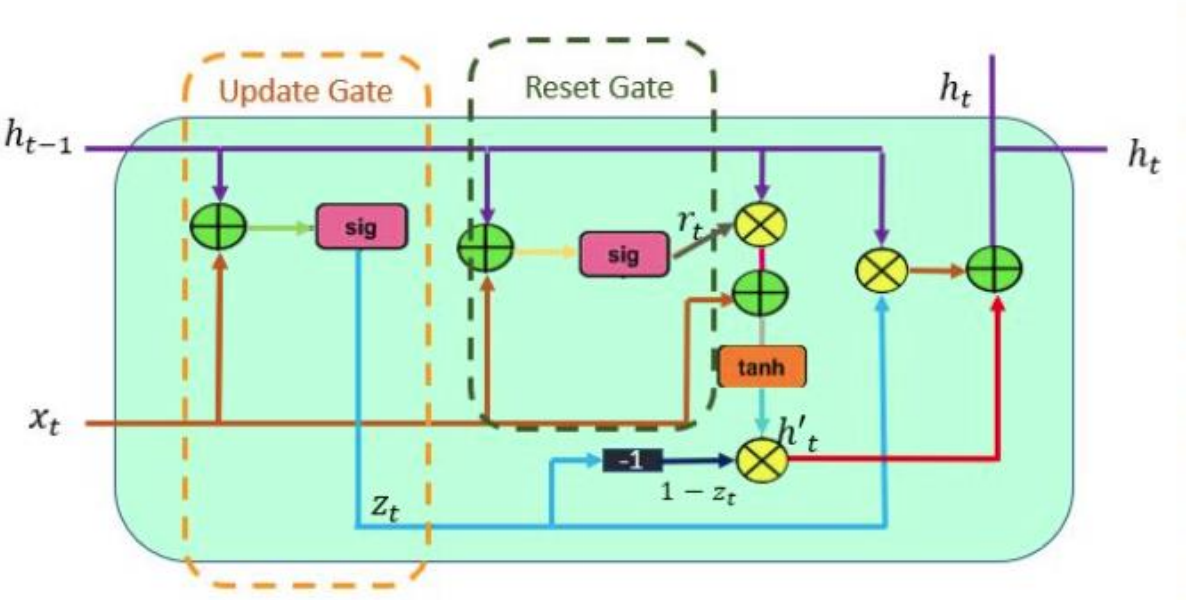
The workflow of the Gated Recurrent Unit, in short GRU, is the same as the RNN but the difference is in the operation and gates associated with each GRU unit. To solve the problem faced by standard RNN, GRU incorporates the two gate operating mechanisms called Update gate and Reset gate.

Update gate

The update gate is responsible for determining the amount of previous information that needs to pass along the next state. This is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient.

Reset gate

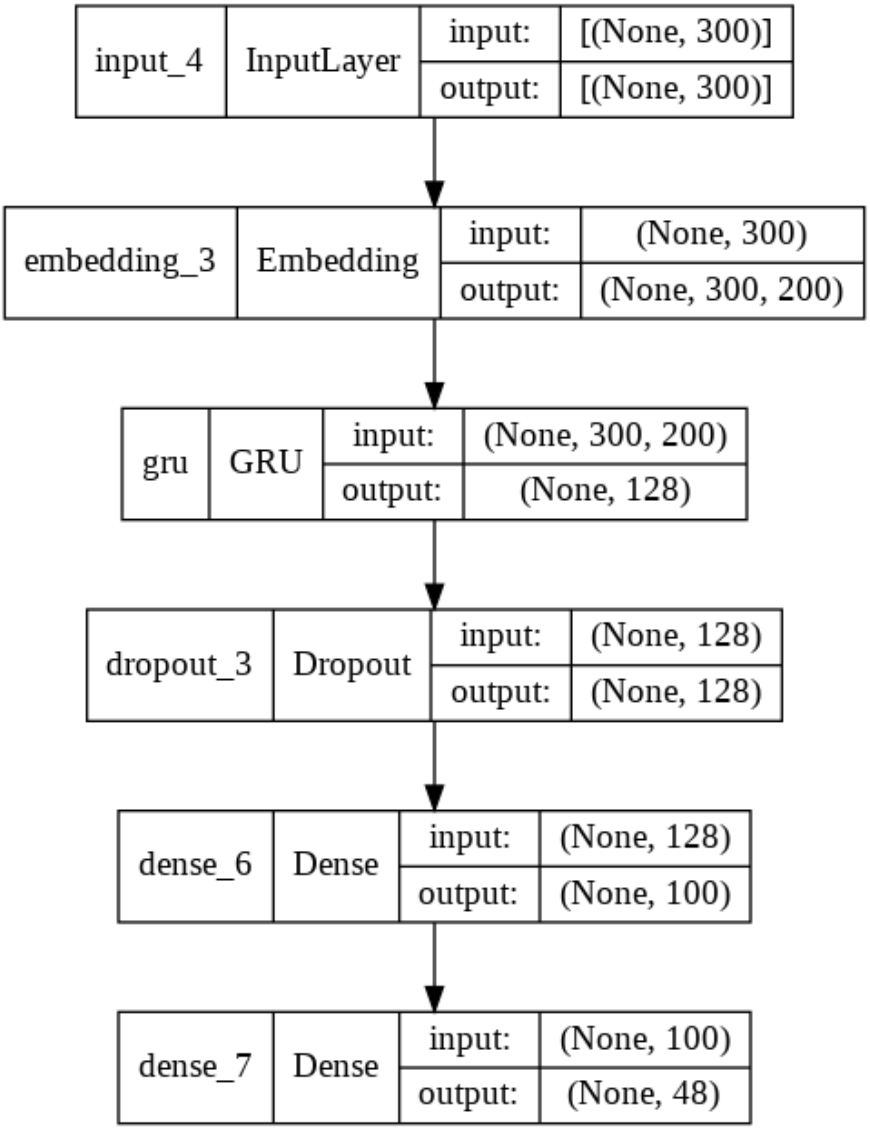
The reset gate is used from the model to decide how much of the past information is needed to neglect; in short, it decides whether the previous cell state is important or not.



GRU

Model Result

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_4 (InputLayer)         [(None, 300)]             0

 embedding_3 (Embedding)      (None, 300, 200)          2340600

 gru (GRU)                    (None, 128)               126720

 dropout_3 (Dropout)          (None, 128)               0

 dense_6 (Dense)              (None, 100)               12900

 dense_7 (Dense)              (None, 48)                4848

=================================================================
Total params: 2,485,068
Trainable params: 2,485,068
Non-trainable params: 0
```


input_4 InputLayer input: [(None, 300)] output: [(None, 300)]
embedding_3 Embedding input: (None, 300) output: (None, 300, 200)
gru GRU input: (None, 300, 200) output: (None, 128)
dropout_3 Dropout input: (None, 128) output: (None, 128)
dense_6 Dense input: (None, 128) output: (None, 100)
dense_7 Dense input: (None, 100) output: (None, 48)


Monitoring the performance of the GRU model
Model Loss
Train
Test
Model Accuracy
Train
Test

MODEL ACCURACY – DEEP LEARNING

Model	accuracy_training	accuracy_test
LSTM	95.51	93.89
Bi-LSTM	95.59	93.56
GRU	95.85	93.99

Model	accuracy_training	accuracy_test
Random Forest	95.99	94.78
Xgboost	89.61	86.82
SVC	94.92	92.33
KNN	91.17	87.40
Naive Bayes	72.65	69.26
LSTM	95.51	93.89
Bi-LSTM	95.59	93.56
GRU	95.85	93.99

Model Tuning

Since Random Forest gave highest accuracy among ML Models, we are hypertuning the Random Forest model

- ✓ Since, Deep learning models have almost same accuracies, we ill fine tune all of them.
- ✓ Try different evaluation parameters
- ✓ Use different hyper parameters - optimizers, loss functions, epochs, learning rate, batch size, checkpointing, early stopping etc. for these models to fine-tune them

HYPER TUNING RANDOM FOREST

HYPER TUNING RANDOM FOREST

```
pipeline = Pipeline([
          ('vect', CountVectorizer()),
          ('tfidf', TfidfTransformer()),
          ('clf', RandomForestClassifier()),
          ])


parameters = {'vect__ngram_range': [(1, 1), (1, 2)],
              'tfidf__use_idf': (True, False),
              'clf__bootstrap': [True],
              'clf__max_depth': [ None, 10,20,30, 40,50],
              'clf__max_features': ['auto', 'sqrt'],
              'clf__min_samples_leaf': [None,1,2, 4, 8,10],
              'clf__n_estimators': [100]}


if __name__ == "__main__":
  grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1, verbose=1,cv=5)
  print("Performing grid search...")
  print("pipeline:", [name for name, _ in pipeline.steps])
  print("parameters:")
  print(parameters)

  RF_CV_Fit = grid_search.fit(X_train, y_train)
  #print("done in %0.3fs" % (time() - t0))
  print()

  print("Best score: %0.3f" % RF_CV_Fit.best_score_)
  print("Best parameters set:")
  best_parameters = RF_CV_Fit.best_estimator_.get_params()
```

```
print("Best score of Random Forest Hyper Tuning using GridSearchCV: %0.3f" % RF_CV_Fit.best_score_)
```

```
Best score of Random Forest Hyper Tuning using GridSearchCV: 0.940
```

+ Code + Text

```
[248] RF_CV_Fit.get_params()
```

```
{'cv': 5,
 'error_score': nan,
 'estimator': Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('clf', RandomForestClassifier())]),
 'estimator__clf': RandomForestClassifier(),
 'estimator__clf__bootstrap': True,
 'estimator__clf__ccp_alpha': 0.0,
 'estimator__clf__class_weight': None,
 'estimator__clf__criterion': 'gini',
 'estimator__clf__max_depth': None,
 'estimator__clf__max_features': 'auto',
 'estimator__clf__max_leaf_nodes': None,
 'estimator__clf__max_samples': None,
 'estimator__clf__min_impurity_decrease': 0.0,
 'estimator__clf__min_samples_leaf': 1,
 'estimator__clf__min_samples_split': 2,
 'estimator__clf__min_weight_fraction_leaf': 0.0,
 'estimator__clf__n_estimators': 100,
 'estimator__clf__n_jobs': None,
 'estimator__clf__oob_score': False,
 'estimator__clf__random_state': None,
 'estimator__clf__verbose': 0,
 'estimator__clf__warm_start': False,
 'estimator__memory': None,
 'estimator__steps': [('vect', CountVectorizer()),
  ('tfidf', TfidfTransformer()),
  ('clf', RandomForestClassifier())],
 'estimator__tfidf': TfidfTransformer(),
```

Hyper tuning Random Forest gives 94% accuracy

HYPER TUNING DEEP LEARNING MODELS

Hyper tuning parameters across LSTM, Bi-LSTM, and GRU

Parameters: Batch size, epochs, number of neurons, dropout

Hyper tuning: Grid Search CV

```
def neural_network(num_neurons=100,act='relu',
                   dropout=0.3,num_class=num_class,maxlen=maxlen,num_words=num_words):

    input_layer = Input(shape=(maxlen,),dtype=tf.int64)
    embed = Embedding(num_words,output_dim=200,input_length=maxlen,weights=[embedding_matrix], trainable=True)(input_layer)
    lstm= LSTM(128)(embed)
    drop=Dropout(dropout)(lstm)
    dense =Dense(num_neurons,activation='relu')(drop)
    out=Dense(num_class,activation='softmax')(dense)
    model_lstm= Model(input_layer,out)
    model_lstm.compile(loss='categorical_crossentropy',optimizer="adam",metrics=['accuracy'])
    return model_lstm

model_lstm = KerasClassifier(build_fn=neural_network,verbose=0)
batch_size = [100]
epochs = [10]
num_neurons = [50,100]
dropout = [0.3]
param_grid = dict(batch_size=batch_size,epochs=epochs,
                  num_neurons=num_neurons,
                  dropout=dropout
                 )
grid = GridSearchCV(estimator=model_lstm,param_grid =param_grid,cv=5,n_jobs=-1)
grid_result_lstm = grid.fit(x_train,ytrain)
grid_result_lstm.best_params_
```

```
{'batch_size': 100, 'dropout': 0.3, 'epochs': 10, 'num_neurons': 100}
```

```
5] print('Best score LSTM GridSearchCV: ',grid_result_lstm.best_score_)
   print('Best param LSTM GridSearchCV: ',grid_result_lstm.best_params_)
   print('Execution time LSTM GridSearchCV: ',grid_result_lstm.refit_time_)
```

Result Hyper tuning

```
print('Best score LSTM GridSearchCV: ',(grid_result_lstm.best_score_)*100)
print('Best score Bi-LSTM GridSerach: ',(grid_result_bi_lstm.best_score_)*100)
print('Best score GRU GridSearchCV: ',(grid_result_gru.best_score_)*100)
print("Best score Random Forest GridSearchCV: ", ( RF_CV_Fit.best_score_)*100)
```

```
Best score LSTM GridSearchCV:  92.13089227676392
Best score Bi-LSTM GridSerach:  91.76104307174683
Best score GRU GridSearchCV:  92.16267585754395
Best score Random Forest GridSearchCV:  94.03951633211271
```

Best score LSTM GridSearchCV: 92.13089227676392
Best score Bi-LSTM GridSerach: 91.76104307174683
Best score GRU GridSearchCV: 92.16267585754395
Best score Random Forest GridSearchCV: 94.03951633211271

1. The GRU model gives a better accuracy among Deep Learning models through Grid Search CV at 92.16%.
2. The Random Forest model gives highest accuracy for Machine Learning models and all models at 94.03%
3. We have also seen Random Forest has higher accuracy, precision and recall in the first cut among all models.

Model Evaluation & Visualization

The best parameters of Random Forest model

```
[275] RF_CV_Fit.best_params_
```

```
{'clf__bootstrap': True,
 'clf__max_depth': None,
 'clf__max_features': 'auto',
 'clf__min_samples_leaf': 1,
 'clf__n_estimators': 100,
 'tfidf__use_idf': False,
 'vect__ngram_range': (1, 2)}
```

```
# Create training and test datasets with 80:20 ratio without augmenatation
X_train, X_test, y_train, y_test = train_test_split(dataset_ML_upsampled.combined_description,
                                                    dataset_ML_upsampled.target,
                                                    test_size=0.20,
                                                    random_state=42)
print('\033[1mShape of the training set:\033[0m', X_train.shape, y_train.shape)
print('\033[1mShape of the test set:\033[0m', X_test.shape, y_test.shape)
```

```
Shape of the training set: (25149,) (25149,)
Shape of the test set: (6288,) (6288,)
```

```
[290] rf_tuned  = RandomForestClassifier(bootstrap= True, max_depth=None, max_features='auto',
                                         min_samples_leaf=1, n_estimators=100)


result = {}   # Create an empty dictionary to later use to store metrics of each of the models

for model, name  in zip([rf_tuned],
                        ['Random Forest - tunes ']):
    result[name] = fit_n_print(model,X_train, X_test, y_train, y_test)
```

Random Forest Hyper Tuned – Classification Report

```
Algorithm: RandomForestClassifier

 Classification report:
              precision    recall  f1-score   support

           0       0.93      0.95      0.94       682
           1       0.87      1.00      0.93       105
           2       0.98      0.88      0.93       117
           3       1.00      1.00      1.00       126
           4       0.98      0.96      0.97       125
           5       0.99      0.95      0.97       117
           6       1.00      1.00      1.00       114
           7       1.00      1.00      1.00       113
           8       0.98      1.00      0.99       127
           9       0.99      1.00      1.00       106
          10       0.98      0.98      0.98       125
          11       0.97      0.87      0.92       131
          12       0.97      0.98      0.97       115
          13       1.00      1.00      1.00       115
          [illegible]

    accuracy                           0.95      6288
   macro avg       0.96      0.95      0.95      6288
weighted avg       0.96      0.95      0.95      6288


 Confusion report:
 [[649   0   0 ...   0   0   0]
 [  0 105   0 ...   0   0   0]
 [  1   0 103 ...   0  11   0]
 ...
 [  3   4   2 ...  61  34   0]
 [  7   0   0 ...   0  93   0]
 [  2   0   0 ...   0   0 113]]
Accuracy Score: 0.9478371501272265
```

Comparison to benchmark

- ✓ The problem statement mentions that around ~25% of incidents are wrongly assigned to functional teams. i.e. 75% are correctly classified.
- ✓ The final Random Forest model gives a accuracy of 94.736%; almost 20% higher than the current benchmark in place.

Implications

- ✓ The accuracy is 94% which is much higher than the current benchmark.
- ✓ The business implication would be more suitable allocation of resources in terms of people leading to reduction of operating costs.
- ✓ Also because of the automation, the customer service will also improve drastically.

Limitations

Technical limitations:

- ✓ The class representation was very skewed in the dataset. More datapoints overcoming this would be ideal.
- ✓ ~40 groups have just 30 or less tickets assigned amongst which 6 groups have just 1 ticket and 4 groups have just 2 tickets each. We had to group classes with <10 datapoints as we couldn't find any patterns for those individual groups given very low datapoints
- ✓ There were more than 30 different languages in the dataset, converting all of them to English didn't work accurately.
- ✓ Also, the timeline of data would be helpful in assessing the frequency in view of recency.

Business limitations:

- ✓ As we combined some of the groups with < 10 data points as we won't be able to assign the tickets for those individual groups. The prediction will be for the combination of those groups. So business might need to restructure their customer service teams accordingly.. We will need more data points for each of those individual groups to be able to assign tickets to these individual groups

Closing Reflection

- ✓ The class imbalance was a major challenge, and we learned data augmentation techniques to try to treat this. We could more variants of augmentation going forward
- ✓ We tried advanced models like LSTM, Bi-LSTM, GRU and got excellent accuracy. So we didn't feel the need to try other advanced models as resource consumption would be higher but scope of accuracy improvement wouldn't be much. Going forward, for some of the other problem statements, we could also try more advanced models such as encoder-decoder, attention models etc.
- ✓ Since, different issues are being raised and in different languages, it would be advisable for the business to create a portal or in the IVR to have drop down menus. This would improve quality of data, reducing variability in data and improve classification.

Appendices:

References:

- https://www.tensorflow.org/
- https://keras.io/
- https://towardsdatascience.com/
- https://elitedatascience.com/
- https://machinelearningmastery.com/
- https://www.greatlearning.in/
- https://medium.com/
- https://stackoverflow.com/
- https://www.kaggle.com/
- https://github.com/
- https://pypi.org/

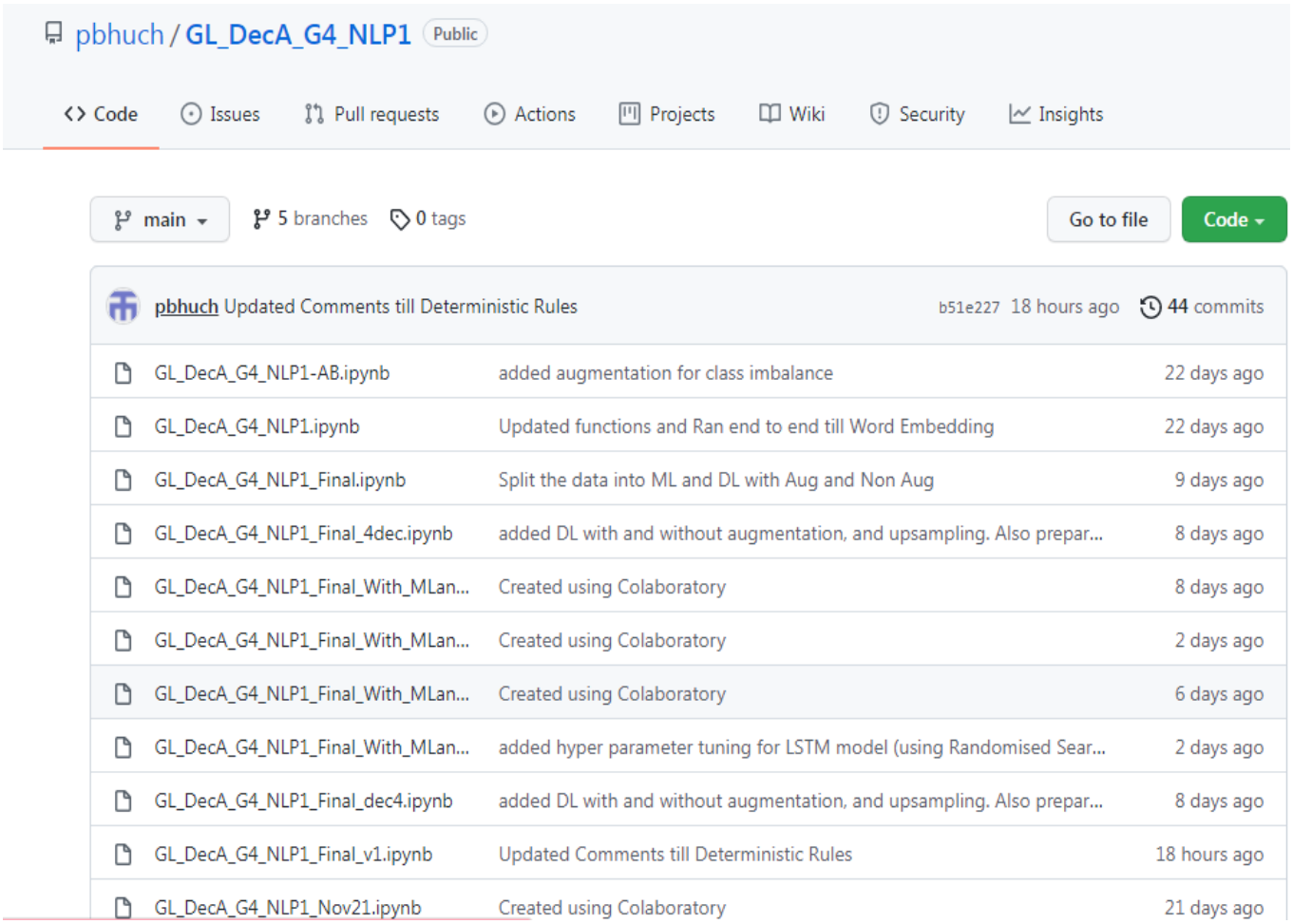
Appendix2 Libraries used:

Sl No	Library/Function	Usage
1.	importmatplotlib.pyplotas plt	For creating the visualization as part of EDA
2.	importseaborn as sns	For visualizing (plotting) the data findings
3.	Importos	This Python's standard utility module is used to interact with this file system, in our case for mounting the google drive and Changing the path to working directory
4.	Importnltk	Natural Language Toolkit provides easy-to-use interface to work with human language data. There are a suite of libraries for classification, tokenization, stemming, tagging, parsing, and Semantic reasoning, wrappers
5.	nltk.download('stopwords')	This is used for identifying and removing the stop-words from the Corpus
6.	nltk.download('wordnet')	This large word database is used to identify and remove the of English Nouns, Adjectives, Adverbs and Verbs from the corpus

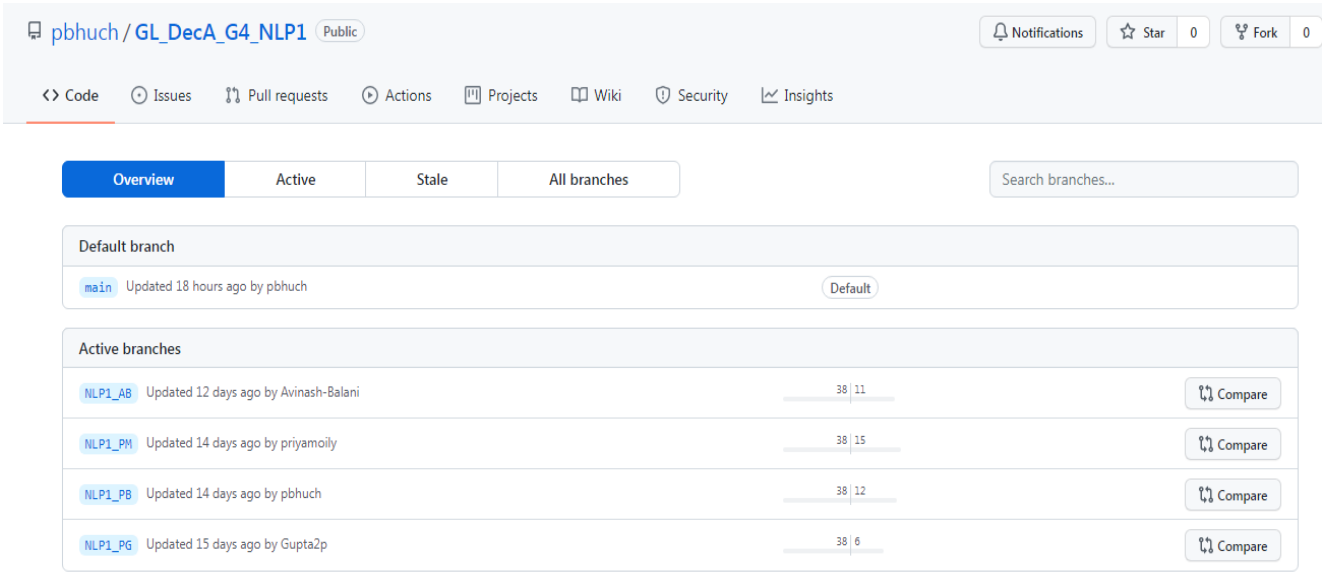
Sl No	Library/Function	Usage
7.	nltk.download('punkt')	This is used to tokenize the sentence
8.	importre	Regex module provides support for regular expressions like search and match
9.	fuzzywuzzy	Fuzzy string matching uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package. This function is used to identify similarity between short description and description, so that a decision whether to Concatenate or take one of the columns
10	Contractions	This function is used to identify and fix contractions such as `you' re`to you `are`

Appendix3 Github usage:

Github, the software development platform has been used here for the collaboration of data, code, and documentation. A private project was created and all the team members have been given access to the project. The project code, project inputs, data and the minutes of meeting, both internal and with the mentor are stored and maintained in the repository.



The tasks assigned to the team is also maintained and managed in the Github by managing different branch For individual task as below:



Issues faced in Github:

Merging individual work into main notebook