2020

Automated Ticket Assignment

SUPERVISED BY: MR. SUMIT KUMAR

SUBMITTED BY: ASHISH ROY, ASHITHA K.R, BRAGADEESH SUNDARARAJAN, PAWAN GUPTA

Contents

1. Team Details

Name	Email Address
Ashish Roy	ashish12459@gmail.com
Ashitha KR	itsashkr@gmail.com
Bragadeesh Sundararajan	bragadeeshs@gmail.com
Pawan Gupta	gupta.pawan227@gmail.com

2. Summary of Problem statement, Data and Findings

Problem Statement

Our problem statement is related to IT incident management. In any Incident Support System which is following Customer Centric Approach, Incident Management plays an important role in delivering quality support to its customers. While an incident report can be logged by anyone in an organization; a right and timely resolution is dependent on area of concern and routing it to the right group of folks with relevant expertise to resolve the same. Thus, in the entire pipeline of incident resolution; getting the ticket assigned as quickly as possible in the right queue plays an important role in the turnaround for closure of ticket it the best possible way. Traditionally all the filed tickets get queued to Service desk team and then it gets manually assigned to the respective teams with right expertise based on the nature of the incident. The Service Desk team (L1/L2) will perform basic analysis on the user's requirement, identify the issue based on given descriptions and assign it to the respective teams.

The manual assignment of these incidents might have below disadvantages:

- More resource usage and expenses.
- Dependency on a specific person/team to initiate the Incident root cause and resolution causing inadvertent delay in initiating assignments; and added cost due to increased turnaround.
- Ineffective use of resources; and limiting resources from expanding their span of influence/skills.
- Human errors - Incidents get assigned to the wrong assignment groups; leading to ineffective utilization of time, skills and resources
- Erroneous assignments leading to need for transfer of tickets from one group to other; wasted efforts, longer time for resolution and poor customer satisfaction

Our Objective:

If this ticket assignment is automated, it will be more cost-effective, optimizing the turnaround for ticket resolution time, enabling the Service Desk team to focus on other productive tasks. Given the problem statement and business value; guided by powerful AI techniques; we would like to build a classifier that can classify the ticket based on analyzing the text in the incident report

Data

We utilized the data shared at the below location as our input data for solving this problem:
https://drive.google.com/open?id=1OZNJm81JXucV3HmZroMq6qCT2m7ez7IJ

3. Overview of the final process

Salient Features of the data

We reviewed the data; and marked below characteristics that were very peculiar to the dataset.

- The dataset had four attributes/columns namely Short Description, Description, Caller and Assignment Group.
- While the above dataset had 8500 records/ticket details; we observed that the data was heavily imbalanced.
- There was a total of 74 assignment groups in the dataset of which Assignment group 0 alone had around 3976 records which attributes to 46.8% of overall data under consideration.; followed by Assignment Group 8 which it had around 661 records which attributed to 7.78%. while there were at least 34 assignment groups that had 20 or less records each.
- We found 8 records which had null values in Short Description column and 1 record which had null value in description.
- There were 250 caller data in the entire dataset. From the static review of data; we observed that caller information did not have much useful data and had a random pattern and thus thought could be eliminated
- The description data did not have any standard format and some of them had snippet of Email/chat dumped
- In addition, text under description also had Symbols and other special characters in the description
- There were also hyperlinks, URLS and few image data references were also found in the description
- Manual inspection also revealed that there were few Spelling mistakes and typo errors too.
- Manual inspection further showed; there were some groups which could be auto assigned based on the pattern; and there were many entries where there was no

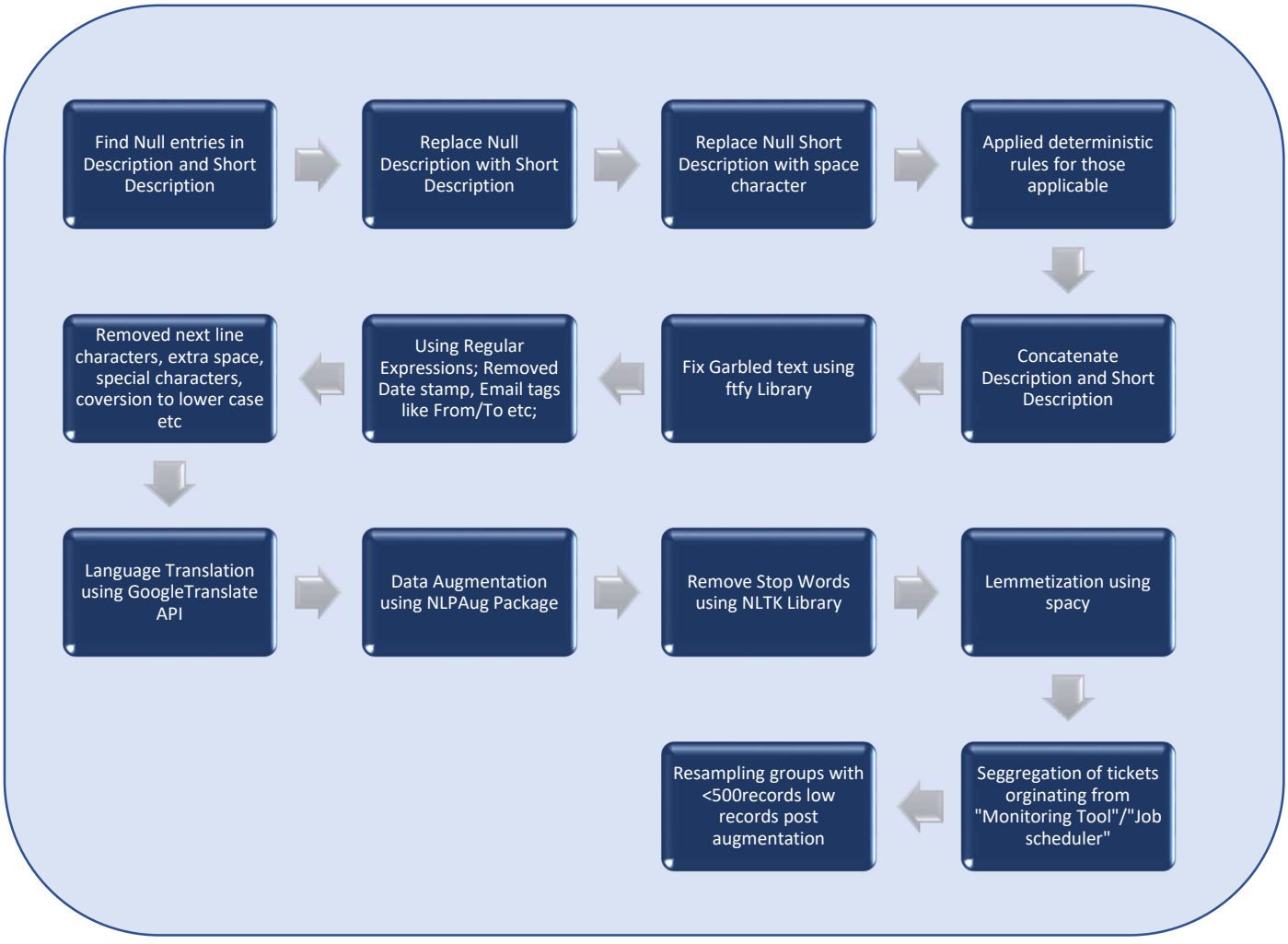
meaningful pattern and made its entry to multiple groups as against assignment to one group.

- In addition to this we found many non-English data too; which was again another aspect that we wanted to consider during data curation

Data Pre-processing Techniques Applied

A High Level Overview

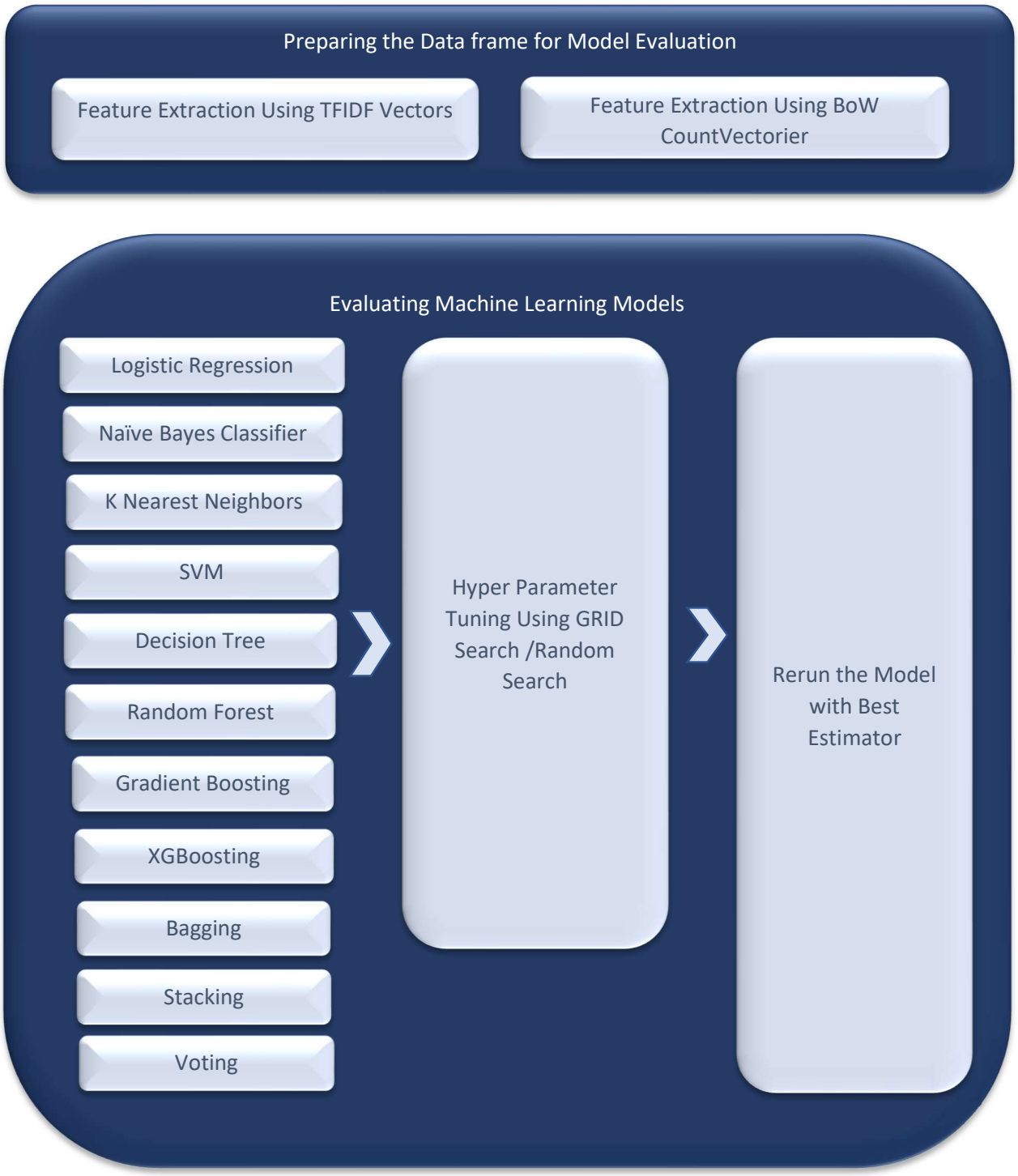
Please find below a high-level overview of data pre-processing techniques applied. We used a combination of visual/manual inspection, data visualization and data analysis to derive each step. We spent a major portion of our capstone time in data analysis and preprocessing; and often found us going back and forth based on the outcomes and observations in the further steps that lead us to revisiting data pre-processing steps and looking for opportunities to further finetune.

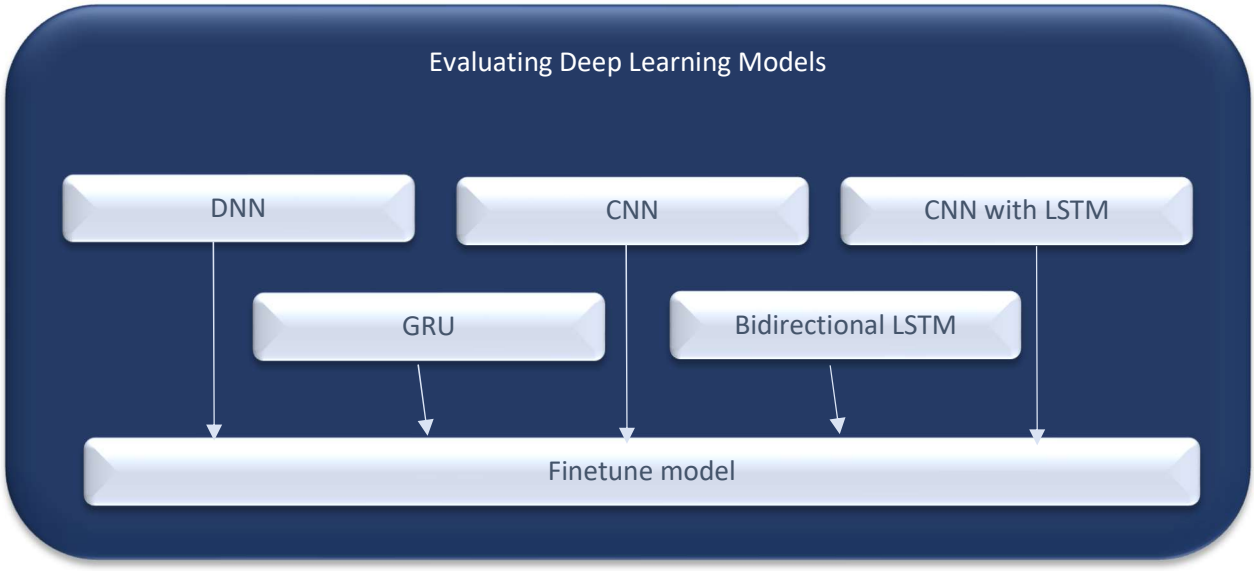


Model Training and Approach Taken

A High Level Overview

Below is a high-level overview of steps we took for model evaluation.





Note: In the first pass; we did feature extraction using bag of words with count vectorizer.
We also explored feature extraction using TFID Vector method to assess relative difference in results. We found that there is ~1% difference in the accuracy and F1 scores by using TFID Vector method as compared to CountVectorizer method. We further evaluated traditional machine learning models and ensemble machine learning techniques.

For Machine model evaluation; though we wished to go through an extensive search using GRIDSearch for all the machine learning models; given the execution time and resource requirements we resorted to Ramdomizedsearch for most of the models and attempted GRIDSearch only for few models

For Deep Learning Models; selected models we fine-tuned adjusting # of hidden layers; dropout and opted for the ones which gave relatively better results

4. Step-by-step walk through the solution

Describe the steps you took to solve the problem. What did you find at each stage, and how did it inform the next steps? Build up to the final solution.

- We mounted the drive and loaded the needed libraries

```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import time, os, sys, itertools, re
from PIL import Image
import warnings, pickle, string
from dateutil import parser
%matplotlib inline

# Data Visualization
import cufflinks as cf
import plotly as py
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot

from ftfy import fix_text, badness

# Traditional Modeling
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, TfidfTransformer
from sklearn.svm import SVC, LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV

# Sequential Modeling
from sklearn import metrics
import tensorflow as tf
tf.config.experimental.list_physical_devices('GPU')
import keras.backend as K
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Concatenate
from tensorflow.keras.layers import Input, Dropout, Flatten, Dense, Embedding, LSTM, GRU, Bidirectional, multiply, Permute
from tensorflow.keras.layers import BatchNormalization, TimeDistributed, Conv1D, MaxPooling1D, Activation, Embedding
from tensorflow.keras.constraints import max_norm, unit_norm
from tensorflow.keras.preprocessing.text import Tokenizer, text_to_word_sequence
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau


# Tools & Evaluation metrics
from sklearn.metrics import confusion_matrix, classification_report, auc
from sklearn.metrics import roc_curve, accuracy_score, precision_recall_curve
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
```

```
Using TensorFlow backend.
```

- Read the data and check details of loaded data frame

From the below it can be seen that there are a total of 8500 records here with 4 attributes of object datatype namely Short Description, Description, Caller and Assignment Group. It can be seen that short description and description field has null values.

```
data=pd.read_excel('/content/drive/MyDrive/Capstone/input_data.xlsx')
#data=pd.read_excel('input_data.xlsx')
data.info()
```

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

- Then we went on to do exploratory Data analysis.

Exploratory Data Analysis

Univariate visualization

Single-variable or univariate visualization is the simplest type of visualization which consists of observations on only a single characteristic or attribute. Univariate visualization includes histogram, bar plots and line charts.

The distribution of Assignment groups

Plots how the assignments groups are scattered across the dataset. The bar chart, histogram and pie chart tells the frequency of any ticket assigned to any group OR the tickets count for each group.

```
data.head()
```

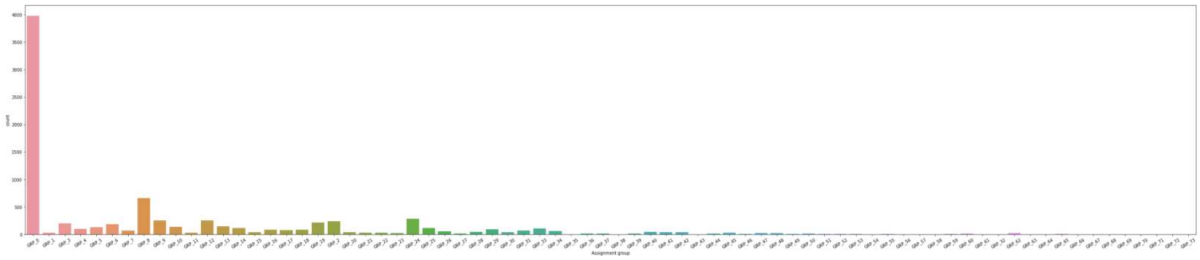
	Short description	Description	Caller	Assignment group
0	login issue	-verified user details.(employee# & manager na...	spxjnwir pjlcoqds	GRP_0
1	outlook	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail...	hmjdrvpb komuaywn	GRP_0
2	cant log in to vpn	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail...	eylqgodm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

```
assignment_group_count=data['Assignment group'].value_counts()
assignment_group_count.describe()
```

```
count      74.000000
mean      114.864865
std       465.747516
min         1.000000
25%         5.250000
50%        26.000000
75%        84.000000
max      3976.000000
Name: Assignment group, dtype: float64
```

- data.head() highlights the top 5 records in the dataframe; and further step indicate we have a total of 74 assignment groups. We can also see that minimum records present in a group is one record and maximum records existing in a group is 3976; and 25% of the groups have 5 or less records so on and so forth. Thus, the data comes across as very skewed. Let us visualize this as below.

```
plt.subplots(figsize=(50,10))
ax=sns.countplot(x='Assignment group', data=data)
ax.set_xticklabels(ax.get_xticklabels(), rotation=30)
plt.tight_layout
plt.show()
```



In [13]:
```
assignment_group_count.head(50)
```

Out[13]:
```
GRP_0     3976
GRP_8      661
GRP_24     289
GRP_12     257
GRP_9      252
GRP_2      241
GRP_19     215
GRP_3      200
GRP_6      184
GRP_13     145
GRP_10     140
GRP_5      129
GRP_14     118
GRP_25     116
GRP_33     107
GRP_4      100
GRP_29      97
GRP_18      88
GRP_16      85
GRP_17      81
GRP_31      69
GRP_7       68
GRP_34      62
GRP_26      56
GRP_40      45
GRP_28      44
GRP_41      40
GRP_15      39
GRP_30      39
GRP_42      37
GRP_20      36
GRP_45      35
GRP_22      31
GRP_1       31
GRP_11      30
GRP_21      29
GRP_47      27
GRP_23      25
GRP_48      25
GRP_62      25
GRP_60      20
GRP_39      19
GRP_27      18
GRP_37      16
GRP_36      15
GRP_44      15
GRP_50      14
GRP_65      11
GRP_53      11
GRP_52       9
Name: Assignment group, dtype: int64
```

- From the plot given the wide range of # of records present in each group is evident in the countplot representation; for enabling readability; we have printed the Top 50 Assignment group count as can be seen above and last 24 assignment group as captured below. You can see

that Group 0 has maximum records (3976) and there are at least 25 groups which has records less than 10.

```
assignment_group_count.tail(24)
```

```
GRP_55    8
GRP_51    8
GRP_59    6
GRP_49    6
GRP_46    6
GRP_43    5
GRP_32    4
GRP_66    4
GRP_68    3
GRP_63    3
GRP_58    3
GRP_38    3
GRP_56    3
GRP_71    2
GRP_57    2
GRP_72    2
GRP_54    2
GRP_69    2
GRP_61    1
GRP_67    1
GRP_70    1
GRP_73    1
GRP_35    1
GRP_64    1
Name: Assignment group, dtype: int64
```

Check Missing Values in dataframe

```
data.isnull().sum()
```

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

- As we observed earlier; above reaffirms that we have 8 records in the dataframe that has null data in short description and 1 record where Description has null data. Let's take closer look at the same as below.
- For one record where description is blank we have copied the short description of that record on to the description; and for the 8 records where we have null values in short description we just replaced that with space character; we have just kept it as such given we plan to concatenate the content of short description on to Description field and use the concatenated text in the further steps.

In [16]:
```
data[data["Short description"].isnull()]
```

Out[16]:

	Short description	Description	Caller	Assignment group
2604	NaN	\r\n\r\nreceived from: ohdrnswl.rezuibdt@gmail...	ohdrnswl rezuibdt	GRP_34
3383	NaN	\r\n-connected to the user system using teamvi...	qftpazns fxpnytmk	GRP_0
3906	NaN	-user unable tologin to vpn.\r\n-connected to...	awpcmsey ctdiuqwe	GRP_0
3910	NaN	-user unable tologin to vpn.\r\n-connected to...	rhwsmefo tvphyura	GRP_0
3915	NaN	-user unable tologin to vpn.\r\n-connected to...	hxripljo efzounig	GRP_0
3921	NaN	-user unable tologin to vpn.\r\n-connected to...	cziadygo veiosxby	GRP_0
3924	NaN	name:wvqgbdhm fwchqjor\nlanguage:\nbrowser:mic...	wvqgbdhm fwchqjor	GRP_0
4341	NaN	\r\n\r\nreceived from: eqmuniov.ehxkcbgj@gmail...	eqmuniov ehxkcbgj	GRP_0

Copy Short Description to Description if the Description value is NaN

In [17]:
```
data.Description.fillna(data["Short description"], inplace = True)
```

In [18]:
```
data[data["Description"].isnull()]
```

Out[18]:

	Short description	Description	Caller	Assignment group

In [19]:
```
data['Short description'] = data['Short description'].replace(np.nan, '', regex=True)
```

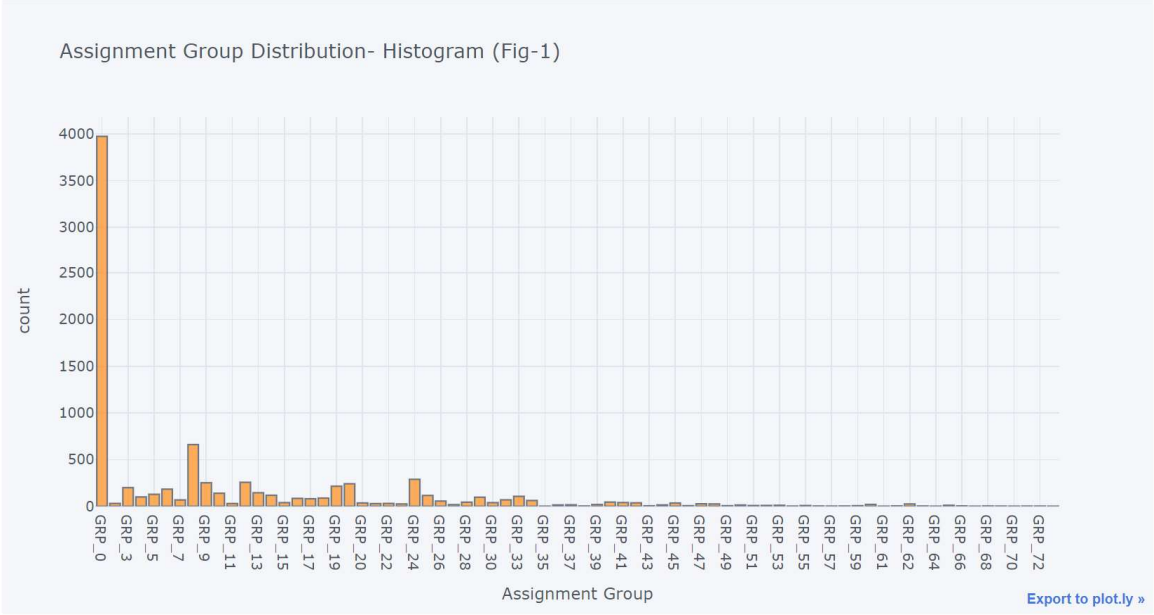
In [20]:
```
data.isnull().sum()
```

Out[20]:
```
Short description    0
Description          0
Caller               0
Assignment group     0
dtype: int64
```

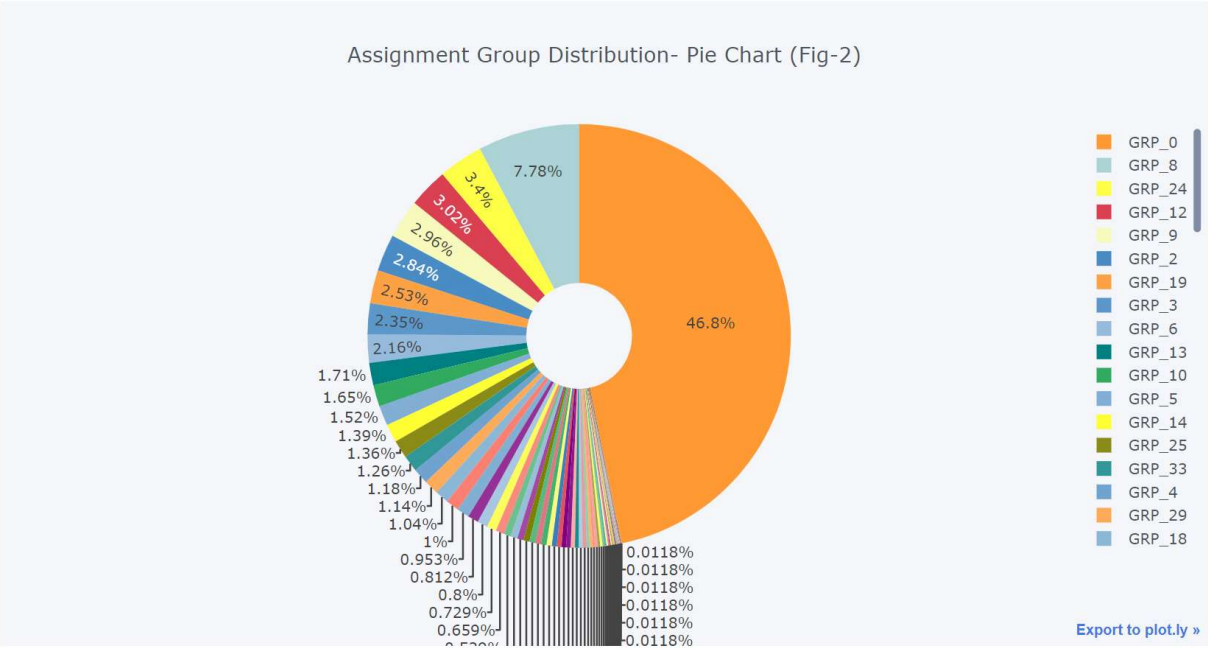
Let's get a histogram view of incidents reported per Assignment group as below

[21]:
```
init_notebook_mode()
cf.go_offline()

# Assignment group distribution
print('\033[1mTotal assignment groups:\033[0m', data['Assignment group'].nunique())

# Histogram
data['Assignment group'].iplot(
    kind='hist',
    xTitle='Assignment Group',
    yTitle='count',
    title='Assignment Group Distribution- Histogram (Fig-1)')

# Pie chart
assgn_grp = pd.DataFrame(data.groupby('Assignment group').size(),columns = ['Count']).reset_index()
assgn_grp.iplot(
    kind='pie',
    labels='Assignment group',
    values='Count',
    title='Assignment Group Distribution- Pie Chart (Fig-2)',
    hoverinfo="label+percent+name", hole=0.25)
```

Total assignment groups: 74

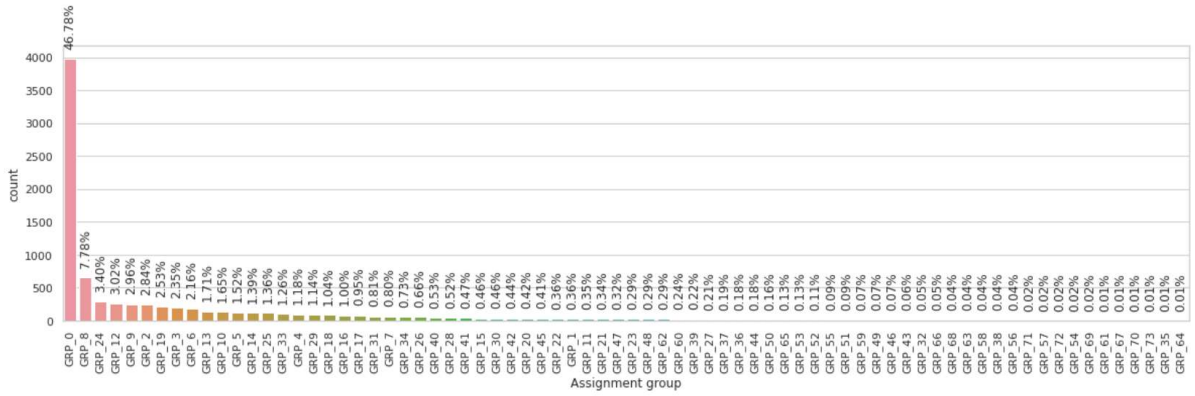


- Further we plotted a pie chart view of the records per assignment group, followed by histogram plot of %ge records falling in each Assignment group as below



Lets visualize the percentage of incidents per assignment group

```
# Plot to visualize the percentage data distribution across different groups
sns.set(style="whitegrid")
plt.figure(figsize=(20,5))
ax = sns.countplot(x="Assignment group", data=data, order=data["Assignment group"].value_counts().index)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
for p in ax.patches:
  ax.annotate(str(format(p.get_height()/len(data.index)*100, '.2f')+"%"), (p.get_x() + p.get_width() / 2., p.get_height()), ha =
'center', va = 'bottom', rotation=90, xytext = (0, 10), textcoords = 'offset points')
```



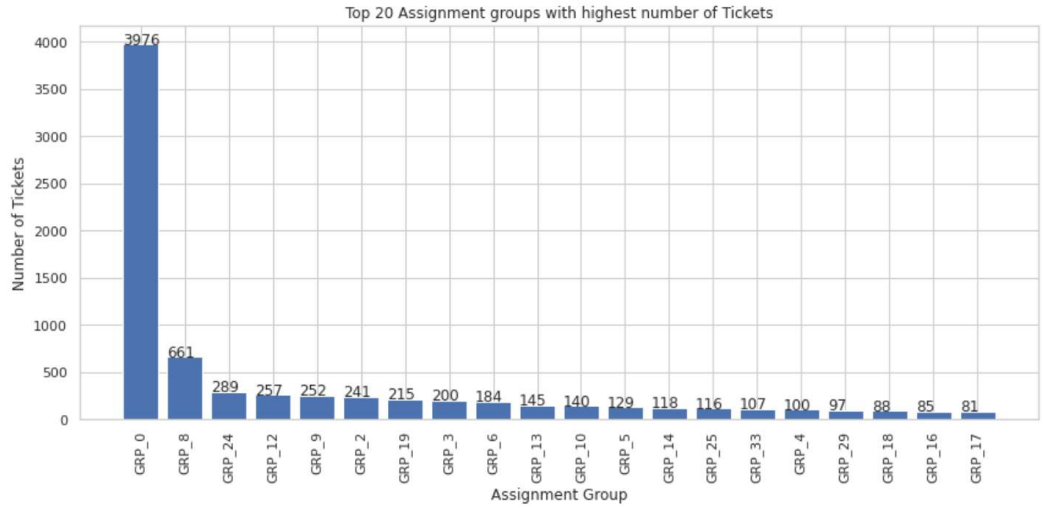
- We could note from the above that 46.8% of records are from assignment group 0; and the second highest number records falling in Group 8 which attributes to close to 7.78%. To get zoomed in view(visualization); we looked at Top 20 and bottom 20 Assignment group as below

greatlearning
Learning for Life

Top 20 and Bottom 20 assignment groups

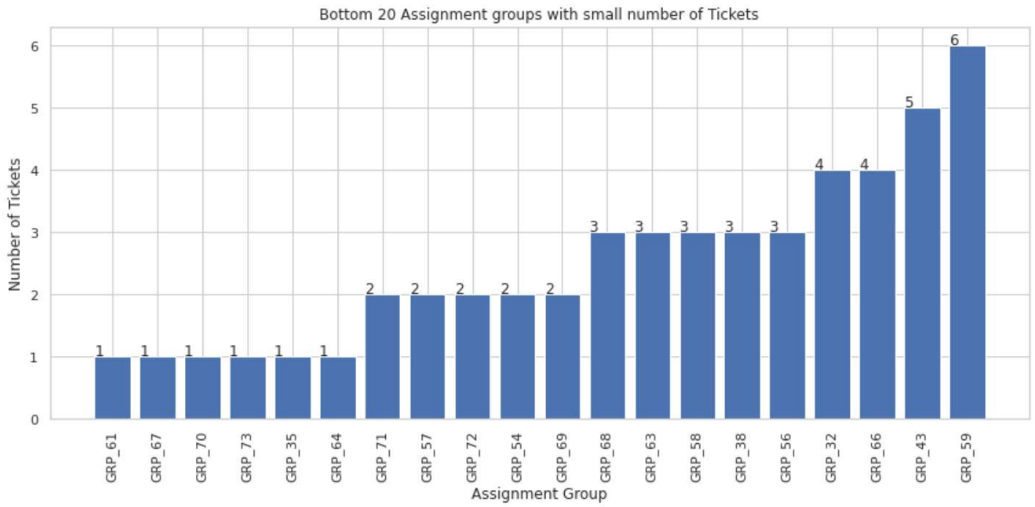
```
top_20 = data['Assignment group'].value_counts().nlargest(20).reset_index()
```

```
plt.figure(figsize=(12,6))
bars = plt.bar(top_20['index'],top_20['Assignment group'])
plt.title('Top 20 Assignment groups with highest number of Tickets')
plt.xlabel('Assignment Group')
plt.xticks(rotation=90)
plt.ylabel('Number of Tickets')

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x(), yval + .005, yval)
plt.tight_layout()
plt.show()
```



```
bottom_20 = data['Assignment group'].value_counts().nsmallest(20).reset_index()
```

```
plt.figure(figsize=(12,6))
bars = plt.bar(bottom_20['index'],bottom_20['Assignment group'])
plt.title('Bottom 20 Assignment groups with small number of Tickets')
plt.xlabel('Assignment Group')
plt.xticks(rotation=90)
plt.ylabel('Number of Tickets')
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x(), yval + .005, yval)
plt.tight_layout()
plt.show()
```



The distribution of Callers

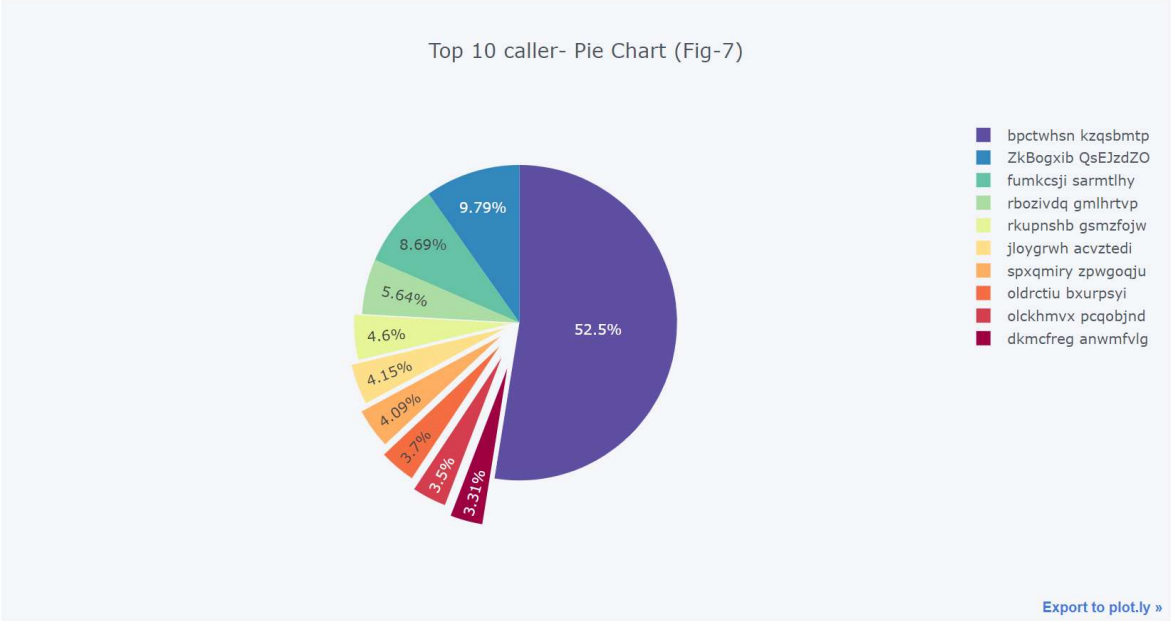
- The above gives a clear view of amount of imbalance in our dataset. Though we observed caller information had some strange string of characters; we also wanted to understand if there is any relevance of the caller information and pattern emerging out of it before deciding to drop that information.

The distribution of Callers

Plots how the callers are associated with tickets and what are the assignment groups they most frequently raise tickets for.

```
# Find out top 10 callers in terms of frequency of raising tickets in the entire dataset
print('\033[1mTotal caller count:\033[0m', data['Caller'].nunique())
df = pd.DataFrame(data.groupby(['Caller']).size().nlargest(10), columns=['Count']).reset_index()
df.iplot(kind='pie',
         labels='Caller',
         values='Count',
         title='Top 10 caller- Pie Chart (Fig-7)',
         colorscale='-spectral',
         pull=[0,0,0,0,0.05,0.1,0.15,0.2,0.25,0.3])
```

Total caller count: 2950



- Above pie chart reflects Top 10 callers who contributed to the incident reporting. Here we learnt that total incidents were reported by 2950 unique callers. Out of the tickets raised by top 10 users; 52.5% of the records seem to be coming from a caller tagged “bpctwhsn kzqsbmtp”. This user contributed to a total of 9.5% incident reporting in the dataset that we have. We manually reviewed the data and observed that there were close to 810 incidents coming from this user; and mostly seem to be coming from an automated system initiated by a monitoring tool. Interestingly this data also showed that it did not have any pattern as such and got assigned to almost 16 different assignment group.

greatlearning
Learning for Life

Top 5 callers in each assignment group

```
top_n = 5
s = data['Caller'].groupby(data['Assignment group']).value_counts()
caller_grp = pd.DataFrame(s.groupby(level=0).nlargest(top_n).reset_index(level=0, drop=True))
caller_grp.head(15)
```

		Caller
Assignment group	**Caller**	
GRP_0	**fumkcsji sarmtlhy**	132
	rbozivdq gmlhrtvp	86
	olckhmvx pcqobjnd	54
	efbwiadp dicafxhv	45
	mfeyouli ndobtzpw	13
GRP_1	**bpctwhsn kzqsbmtp**	6
	jloygrwh acvztedi	4
	jyoqwxhz clhxsoqy	3
	spxqmiry zpwgoqju	3
	kbnfxpsy gehxzayq	2
GRP_10	**bpctwhsn kzqsbmtp**	60
	ihfkwzjd erbxoyqk	6
	dizquolf hlykecxa	5
	gnasmtvx cwxtsvkm	3
	hlrmufzx qcdzierm	3

- In the above step we attempted to see Top 5 callers in the assignment_group and printed first 15 records basically covering three groups.
- We further wanted to understand what is the variation in the # of words present in the incident ; as well as the total length of the incident report; (as we observed during manual inspection that there were many records which had content that looked like a dump of an email to a short one liner failure notification.)

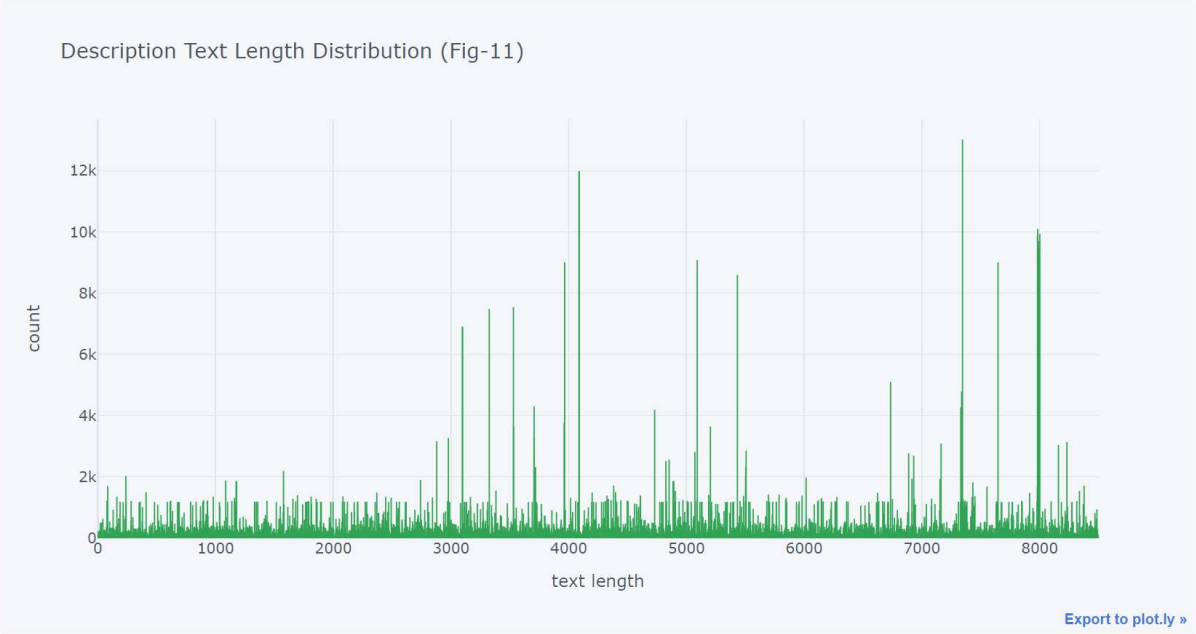
The distribution of description lengths

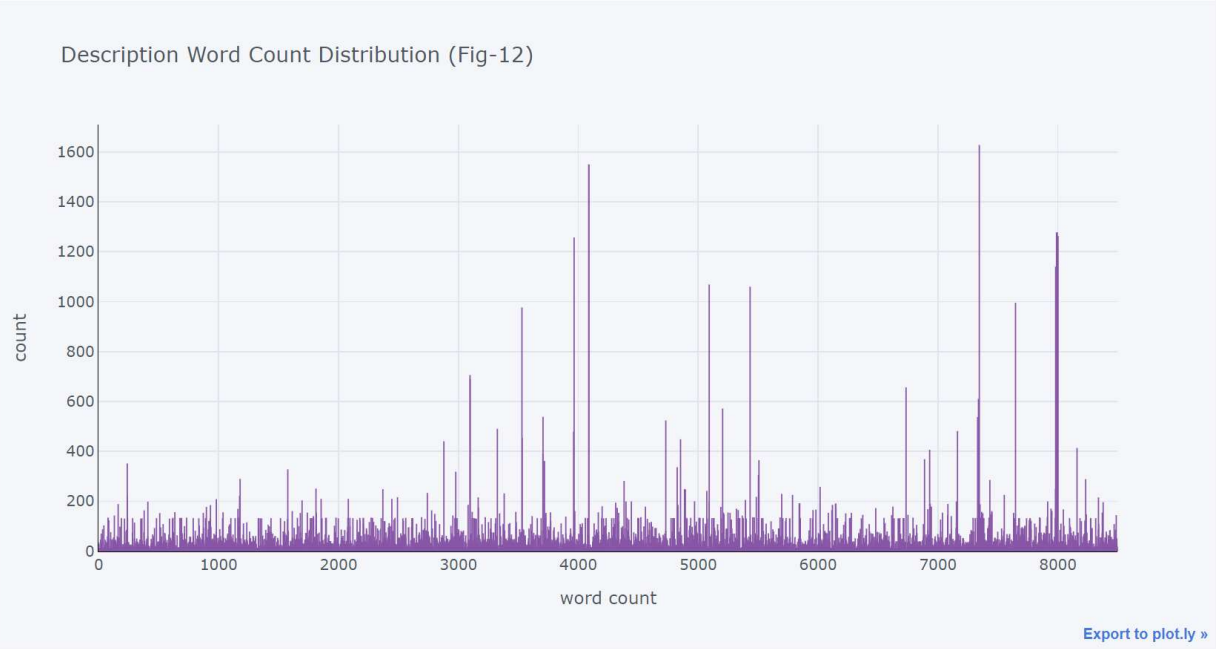
Plots the variation of length and word count of new description attribute

```
data.insert(1, 'desc_len', data['Description'].astype(str).apply(len))
data.insert(5, 'desc_word_count', data['Description'].apply(lambda x: len(str(x).split())))
data.head()
```

	Short description	desc_len	Description	Caller	Assignment group	desc_word_count
0	login issue	206	-verified user details.(employee# & manager na...	spxjnwir pjlcoqds	GRP_0	33
1	outlook	194	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail...	hmjdrvpb komuaywn	GRP_0	25
2	cant log in to vpn	87	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail...	eylqgodm ybqkwiam	GRP_0	11
3	unable to access hr_tool page	29	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0	5
4	skype error	12	skype error	owlgqjme qhcozdfx	GRP_0	2

```
# Description text length
data['desc_len'].iplot(
    kind='bar',
    xTitle='text length',
    yTitle='count',
    colorscale='-ylgn',
    title='Description Text Length Distribution (Fig-11)')

# Description word count
data['desc_word_count'].iplot(
    kind='bar',
    xTitle='word count',
    linecolor='black',
    yTitle='count',
    colorscale='-bupu',
    title='Description Word Count Distribution (Fig-12)')
```





It could be observed that the description text length and word count has too much of variation.

- As a next step; we established deterministic rule for cases where it is straight forward assignment and do not require machine learning/DL workflows. This is primarily derived based on the manual inspection findings; and created as a function of specific patterns observed between Short Description, Description and group allocated. As an example, if the short description had word ‘erp’ and ‘EU_tool’ both; then the group that particular incident needs to get assigned is GRP_25. Another example is , if ‘finance_app’ is there in short description or Description AND Hostname_1132 is not in short description, the incident need to get categorized as GRP_55.

greatlearning
Learning for Life

Create a rule based engine

```
df_rules = pd.read_csv('/content/drive/MyDrive/Capstone/Rule_matrix.csv')
#df_rules = pd.read_csv("Rule_matrix.csv")
```

```
def applyRules(datadf,rulesdf,Description,ShortDescription):
    datadf['pred_group'] = np.nan
    for i, row in rulesdf.iterrows():
        for j, row in datadf.iterrows():
            if pd.notna(datadf[ShortDescription][j]):
                if (('erp' in datadf[ShortDescription][j]) and (('EU_tool' in datadf[ShortDescription][j]))):
                        datadf['pred_group'][j] = 'GRP_25'
        for j, row in datadf.iterrows():
            if pd.notna(datadf[Description][j]):
                if (datadf[Description][j] == 'the'):
                    datadf['pred_group'][j] = 'GRP_17'

                if (('finance_app' in ((datadf[ShortDescription][j]) or datadf[Description][j])) and ('HostName_1132' not in dat
adf[ShortDescription][j])):
                    datadf['pred_group'][j] = 'GRP_55'

                if (('processor' in datadf[Description][j]) and ('engg' in datadf[Description][j])):
                    datadf['pred_group'][j] = 'GRP_58'


        if rulesdf['Short Desc Rule'][i] == 'begins with' and rulesdf['Desc Rule'][i] == 'begins with' and pd.isna(rulesdf['Use
r'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[ShortDescription][j]) and pd.notna(datadf[Description][j]):
                    if ((datadf[ShortDescription][j].startswith(rulesdf['Short Dec Keyword'][i])) and (datadf[Description][j].st
artswith(rulesdf['Dec keyword'][i]))):
                        datadf['pred_group'][j] = rulesdf['Group'][i]

        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'begins with' and pd.notna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]) and pd.notna(datadf['Caller'][j]):
                    if ((datadf[Description][j].startswith(rulesdf['Desc Rule'][i]) and (rulesdf['User'][i] == datadf['Caller']
[j]))):
                        datadf['pred_group'][j] = rulesdf['Group'][i]

        if rulesdf['Short Desc Rule'][i] == 'contains' and pd.notna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if (pd.notna(datadf[ShortDescription][j]) and pd.notna(datadf['Caller'][j])):
                     if ((rulesdf['Short Dec Keyword'][i] in datadf[ShortDescription][j]) and (rulesdf['User'][i] == datadf['Cal
ler'][j])):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
        if rulesdf['Short Desc Rule'][i] == 'contains' and pd.isna(rulesdf['Desc Rule'][i]) and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[ShortDescription][j]):
                    if (rulesdf['Short Dec Keyword'][i] in datadf[ShortDescription][j]):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'begins with' and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]):
                    if (datadf[Description][j].startswith(rulesdf['Dec keyword'][i])):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
```

```
        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'contains' and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]):
                    if (rulesdf['Dec keyword'][i] in datadf[Description][j]):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'not contain' and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]):
                    if (rulesdf['Dec keyword'][i] in datadf[Description][j]):
                        datadf['pred_group'][j] = rulesdf['Group'][i]


        if rulesdf['Short Desc Rule'][i] == 'not contain' and pd.isna(rulesdf['Desc Rule'][i]) and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():

                if pd.notna(datadf[ShortDescription][j]):
                    if (rulesdf['Short Dec Keyword'][i] in datadf[ShortDescription][j]):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'not contain' and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]):
                    if (datadf[Description][j].startswith(rulesdf['Dec keyword'][i])):
                        datadf['pred_group'][j] = rulesdf['Group'][i]
        if pd.isna(rulesdf['Short Desc Rule'][i]) and rulesdf['Desc Rule'][i] == 'contains' and pd.isna(rulesdf['User'][i]):
            for j, row in datadf.iterrows():
                if pd.notna(datadf[Description][j]):
                    if (rulesdf['Dec keyword'][i] in datadf[Description][j]):
                        datadf['pred_group'][j] = rulesdf['Group'][i]

    return datadf
```

```
rules_applied_df = applyRules(data,df_rules,'Description','Short description')
rules_applied_df
```

	Short description	desc_len	Description	Caller	Assignment group	desc_word_count	pred_group
0	login issue	206	-verified user details.(employee# & manager na...	spxjnwir pjlcoqds	GRP_0	33	NaN
1	outlook	194	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail...	hmjdrvpb komuaywn	GRP_0	25	NaN
2	cant log in to vpn	87	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail...	eylqgodm ybqkwiam	GRP_0	11	NaN
3	unable to access hr_tool page	29	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0	5	NaN
4	skype error	12	skype error	owlgqjme qhcozdfx	GRP_0	2	NaN
...	...	...	...	...	...	...	...
8495	emails not coming in from zz mail	141	\r\n\r\nreceived from: avglmrts.vhqmtiua@gmail...	avglmrts vhqmtiua	GRP_29	19	NaN
8496	telephony_software issue	24	telephony_software issue	rbozivdq gmlhrtvp	GRP_0	2	NaN
8497	vip2: windows password reset for tifpdchb pedx...	50	vip2: windows password reset for tifpdchb pedx...	oybwdsgx oxyhwrfz	GRP_0	7	NaN
8498	machine nÃ£o estÃ¡ funcionando	103	i am unable to access the machine utilities to...	ufawcgob aowhxjky	GRP_62	17	NaN
8499	an mehreren pc`s lassen sich verschiedene prgr...	82	an mehreren pc`s lassen sich verschiedene prgr...	kqvbrspl jyzoklfx	GRP_49	11	NaN

8500 rows × 7 columns

```
rules_applied_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8500 entries, 0 to 8499
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Short description  8500 non-null   object
 1   desc_len           8500 non-null   int64 
 2   Description        8500 non-null   object
 3   Caller             8500 non-null   object
 4   Assignment group   8500 non-null   object
 5   desc_word_count    8500 non-null   int64 
 6   pred_group         301 non-null    object
dtypes: int64(2), object(5)
memory usage: 465.0+ KB
```

- Post applying this method on the dataframe; we looked at how many cases could be handled by deterministic rule. For this we created a column named pred_group which held the group categorization if the rule engine could find a specific group to which the incident could be allocated. From the above you can see that around 200+ incidents could be handled by deterministic rule function that we created.

```
rules_applied_df = rules_applied_df[(rules_applied_df['pred_group'].isna())]
rules_applied_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8199 entries, 0 to 8499
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Short description  8199 non-null   object
 1   desc_len           8199 non-null   int64 
 2   Description        8199 non-null   object
 3   Caller             8199 non-null   object
 4   Assignment group   8199 non-null   object
 5   desc_word_count    8199 non-null   int64 
 6   pred_group         0 non-null      object
dtypes: int64(2), object(5)
memory usage: 512.4+ KB
```

```
assignment_group_count=rules_applied_df['Assignment group'].value_counts()
assignment_group_count.describe()
```

```
count      62.000000
mean      132.241935
std       488.873469
min         1.000000
25%        12.250000
50%        33.000000
75%        99.250000
max      3833.000000
Name: Assignment group, dtype: float64
```

- From the above you can see that there are 8199 records for which classification could not be done via rule engine. And these 8199 records are seen to be scattered across 62 Assignment groups.

- We went ahead to concatenate Short description and Description column and created a New Description column which is what we will use for further problem solving. Since content from Description column and short description column is already folded into New Description column we can eliminate short description column, description column and other columns that we temporarily created for data analysis and visualization.

Concatenate Short Description and Description Column into New Description, drop the previous columns

```
#Concatenate Short Description and Description columns
rules_applied_df['New Description'] = rules_applied_df['Description'] + ' ' +rules_applied_df['Short description']

clean_data=rules_applied_df.drop(['Short description', 'Description', 'pred_group', 'desc_len', 'desc_word_count'], axis=1)
```

```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8199 entries, 0 to 8499
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Caller            8199 non-null   object
 1   Assignment group  8199 non-null   object
 2   New Description   8199 non-null   object
dtypes: object(3)
memory usage: 256.2+ KB
```

- As the next step; we wanted to fix some Garbled text that was observed in the input dataframe; and we used ftfy library for the same as below. ftfy Library is known to take bad Unicode and output good Unicode, for use in Unicode-aware code

Fixing Garbled Text/ Mojibake using ftfy library ¶

In [39]:
```
# Write a function to apply to the dataset to detect Mojibakes
def is_mojibake_impacted(text):
    if not badness.sequence_weirdness(text):
        # nothing weird, should be okay
        return True
    try:
        text.encode('sloppy-windows-1252')
    except UnicodeEncodeError:
        # Not CP-1252 encodable, probably fine
        return True
    else:
        # Encodable as CP-1252, Mojibake alert level high
        return False
# Check the dataset for mojibake impact
clean_data[~clean_data.iloc[:,:].applymap(is_mojibake_impacted).all(1)]
```

Out[39]:

	Caller	Assignment group	New Description
99	ecprjbod litmjwsy	GRP_0	\n\nreceived from: ecprjbod.litmjwsy@gmail.com...
116	bgqpotek cuxakvml	GRP_0	\r\n\r\nreceived from: bgqpotek.cuxakvml@gmail...
124	tvcdfqgp nrbcqwgj	GRP_0	from: tvcdfqgp nrbcqwgj \nsent: friday, octobe...
164	tycludks cjofwigv	GRP_0	\n\nreceived from: abcdri@company.com\n\nwindy...
170	fbvpcytz nokypgvx	GRP_18	\n\nreceived from: fbvpcytz.nokypgvx@gmail.com...
...	...	...	...
8470	azxhejvq fyemlavd	GRP_16	from: mikhghytr wafglhdrhjop \nsent: thursday,...
8471	xqyjztnm onfusvlz	GRP_30	to å°□è´ºï¼Œæ—©ä¸Šç”µè„‘å¼€æœºå¼€ä¸□å‡ºæ□¥ ç”µ...
8480	nlearzwi ukdzstwi	GRP_9	\r\n\r\nreceived from: nlearzwi.ukdzstwi@gmail...
8498	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...
8499	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...

820 rows × 3 columns

In [40]:
```
# Take an example of row# 8471 Short Desc and fix it
print('Grabled text: \033[1m%s\033[0m\nFixed text: \033[1m%s\033[0m' % (clean_data['New Description'][8471],
                                                                       fix_text(clean_data['New Description'][8471])))

# List all mojibakes defined in ftfy library
print('\nMojibake Symbol RegEx:\n', badness.MOJIBAKE_SYMBOL_RE.pattern)
```

```
Grabled text: to å°□è´ºï¼Œæ—©ä¸Šç”µè„‘å¼€æœºå¼€ä¸□å‡ºæ□¥ ç”µè„‘å¼€æœºå¼€ä¸□å‡ºæ□¥
Fixed text: to 小贺,早上电脑开机开不出来 电脑开机开不出来

Mojibake Symbol RegEx:
 [ÀÃÎÏÐÑØÙÄÕÐÑŘÙ][□-□€ƒ„†‡ˆ‰‹Œ‘•˜œŸ¡¢£¤¥¦§¨ª«¬¯°±²³µ¶·¸¹º»¼½¾¿ˇ˘˝]|[ÂÃÎÏÐÑØÙÄÕÐÑŘÙ][›»‘’“”´®™]\w|×[□-□ƒ„†‡ˆ‰‹Œ‘•˜œŸ¡¦§¨ª«¬¯°²³ˇ
ˇ›‘’“”´®™]|[¬√][ÄÅÇÉÑÖÜáàâäãåçéèêëíìîïñúùûü†¢£§¶ß®©™≠ÆØ¥ªæø≤≥]|\w√[±∂]\w|◊|[ðđ][Ÿ□]|â€|вђ[љћ¦°№™Ќ“”]
```

```
# Sanitize the dataset from Mojibakes
clean_data['New Description'] = clean_data['New Description'].apply(fix_text)

# Visualize that row# 8471
clean_data.loc[8471]
```

```
Caller                      xqyjztnm onfusvlz
Assignment group                    GRP_30
New Description     to 小贺,早上电脑开机开不出来 电脑开机开不出来
Name: 8471, dtype: object
```

- Then we went on to do further data processing. One of the primary/initial step that we identified needing attention was to take care of specific challenges that we observed in the data such as date, email tags like To, From, subject, Received from, sent, cc, bcc, html tags, special

characters which had to be eliminated as it was not providing any relevant information that needed to be retained for training. We also needed to get all the text in lower case in order to ensure that case sensitivity doesn't impact the learning process. In order to accomplish this, we used regular expressions in python

Cleaning & Processing the data

```
def date_validity(date_str):
    try:
        parser.parse(date_str)
        return True
    except:
        return False
```

```
def process(text_string):
    text=text_string.lower()
    text_string = ' '.join([w for w in text_string.split() if not date_validity(w)])
    text_string = re.sub(r"received from:",'',text_string)
    text_string = re.sub(r"from:",' ',text_string)
    text_string = re.sub(r"to:",' ',text_string)
    text_string = re.sub(r"subject:",' ',text_string)
    text_string = re.sub(r"sent:",' ',text_string)
    text_string = re.sub(r"ic:",' ',text_string)
    text_string = re.sub(r"cc:",' ',text_string)
    text_string = re.sub(r"bcc:",' ',text_string)
    text_string = re.sub(r'\S*@\S*\s?', '', text_string)
    text_string = re.sub(r'\d+','' ,text_string)
    text_string = re.sub(r'\n',' ',text_string)
    text_string = re.sub(r'#','', text_string)
    text_string = re.sub(r'&;?', 'and',text_string)
    text_string = re.sub(r'\&\w*;', '', text_string)
    text_string = re.sub(r'https?:\/\/.*\/\w*', '', text_string)
    text_string= ''.join(c for c in text_string if c <= '\uFFFF')
    text_string = text_string.strip()
    text_string = re.sub(r"\s+[a-zA-Z]\s+", ' ', text_string)
    text_string = re.sub(' +', ' ', text_string)
    text_string = text_string.replace(r'\b\w\b','').replace(r'\s+', ' ')
    text_string = text_string.strip()
    return text_string
```

```
clean_data["Clean_Description"] = clean_data["New Description"].apply(process)
```

- After performing these steps, we looked at the dataframe again and confirmed that intended operations have successfully completed. As you can note in the below snapshot the Clean_description column has text where email tags are removed; special characters are removed or replaced with designated string such as & was replaced with "and" etc.

```
clean_data
```

	Caller	Assignment group	New Description	Clean_Description
0	spxjnwir pjlcoqds	GRP_0	-verified user details.(employee# & manager na...	-verified user details.(employee and manager n...
1	hmjdrvpb komuaywn	GRP_0	\n\nreceived from: hmjdrvpb.komuaywn@gmail.com...	hello team, my meetings/skype meetings etc are...
2	eylqgodm ybqkwiam	GRP_0	\n\nreceived from: eylqgodm.ybqkwiam@gmail.com...	hi cannot log on to vpn best cant log in to vpn
3	xbkucsvz gcpydteq	GRP_0	unable to access hr_tool page unable to access...	unable to access hr_tool page unable to access...
4	owlgqjme qhcozdfx	GRP_0	skype error skype error	skype error skype error
...	...	...	...	...
8495	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...
8496	rbozivdq gmlhrtvp	GRP_0	telephony_software issue telephony_software issue	telephony_software issue telephony_software issue
8497	oybwdsgx oxyhwrfz	GRP_0	vip2: windows password reset for tifpdchb pedx...	vip: windows password reset for tifpdchb pedxr...
8498	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...
8499	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...

8199 rows × 4 columns

- As a next step we went on to detect presence of various languages in the dataframe. We used langdetect library; which is a module that is part of Google language detection library which supports 50 plus languages. Since this doesn't come by default with python ; we installed the same.

Language Translation ¶

Dataset has many language text, prominently German and Chinese. See below chart.

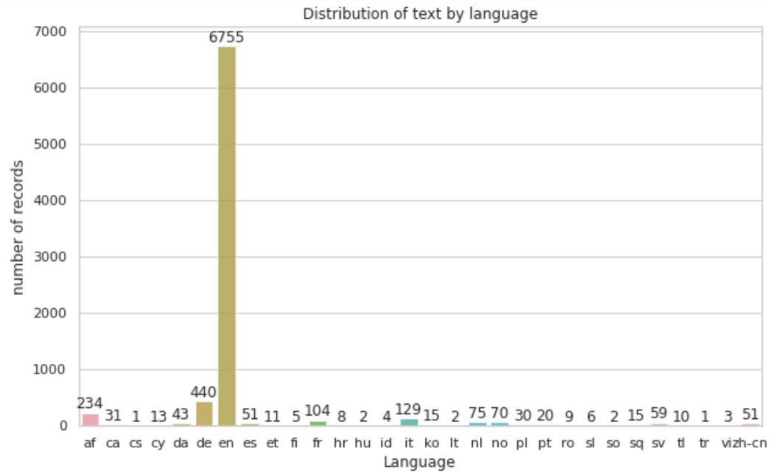
```
: !pip install langdetect
```

```
Collecting langdetect
  Downloading https://files.pythonhosted.org/packages/56/a3/8407c1e62d5980188b4acc45ef3d94b933d14a2ebc9ef3505f22cf772570/langdet
ect-1.0.8.tar.gz (981kB)
     |████████████████████████████████| 983kB 23.2MB/s
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from langdetect) (1.15.0)
Building wheels for collected packages: langdetect
  Building wheel for langdetect (setup.py) ... done
  Created wheel for langdetect: filename=langdetect-1.0.8-cp36-none-any.whl size=993194 sha256=12e312795796a87356d8c999d6c665c33
bd17158356adecf176c18ac4d966a6a
  Stored in directory: /root/.cache/pip/wheels/8d/b3/aa/6d99de9f3841d7d3d40a60ea06e6d669e8e5012e6c8b947a57
Successfully built langdetect
Installing collected packages: langdetect
Successfully installed langdetect-1.0.8
```

```
: from langdetect import detect

def fn_lang_detect(df):
   try:
      return detect(df)
   except:
      return 'no'

clean_data['language'] = clean_data['Clean_Description'].apply(fn_lang_detect)
```

```
x = clean_data["language"].value_counts()
x=x.sort_index()
plt.figure(figsize=(10,6))
ax= sns.barplot(x.index, x.values, alpha=0.8)
plt.title("Distribution of text by language")
plt.ylabel('number of records')
plt.xlabel('Language')
rects = ax.patches
labels = x.values
for rect, label in zip(rects, labels):
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width()/2, height + 5, label, ha='center', va='bottom')
plt.show();
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and pa
ssing other arguments without an explicit keyword will result in an error or misinterpretation.
```



We are using the google translate api's for Language conversion. But using library packages such as Goslate, there is restrictions to convert number of records. Due to this we have used Googletranslate functionality in google sheets for Language conversion. https://support.google.com/docs/answer/3093331?hl=en

- After running the input data through langdetect; it returned distribution of identified languages as above. We observed that the predominant language preset other than English was German and Chinese. So initially we focused mainly on translating German and Chineese language back to English; due to two factors. 1. Manual inspection of langdetect output in some cases it was observed that the detection was incorrect. 2. We were using Goslate for translation and it was allowing us to convert only 100 records every day due to restrictions on free usage of google translation API. However, we recognized during model tuning that its critical to get remaining languages also translated back to English to fine tune classification accuracy. Thus, we explored further and resorted to using google sheets. We imported the data into google sheets and applied a pre-existing formula ["=GOOGLETRANSLATE(**cell** with **text**, "source **language**", "target **language**")] using "GOOGLETRANSLATE" function for all the rows in the data frame . For identifying source language, we used "auto" parameter which was more accurate than langdetect. This was done offline and saved as pkl file.

Loading the consolidated final translated pickle file which contains the language translations done offline.

```
with open('/content/drive/MyDrive/Capstone/final_translated.pkl','rb') as f:
#with open('final_translated.pkl','rb') as f:
    clean_data = pickle.load(f)
```

```
clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8199 entries, 0 to 8198
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Caller             8199 non-null   object
 1   Assignment group   8199 non-null   object
 2   New Description    8199 non-null   object
 3   Clean_Description  8199 non-null   object
 4   language           8199 non-null   object
 5   Translated_Text    8199 non-null   object
dtypes: object(6)
memory usage: 384.5+ KB
```

```
#check the dataframe
clean_data.tail()
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...
8195	rbozivdq gmlhrtvp	GRP_0	telephony_software issue telephony_software issue	telephony_software issue telephony_software issue	en	telephony_software issue telephony_software issue
8196	oybwdsgx oxyhwrfz	GRP_0	vip2: windows password reset for tifpdchb pedx...	vip: windows password reset for tifpdchb pedxr...	en	vip: windows password reset for tifpdchb pedxr...
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...

```
assignment_group_cnt=clean_data['Assignment group'].value_counts()
assignment_group_cnt.describe()
```

```
count      62.000000
mean      132.241935
std       488.873469
min         1.000000
25%        12.250000
50%        33.000000
75%        99.250000
max      3833.000000
Name: Assignment group, dtype: float64
```

- In the above step we confirmed that we still have the remaining 62 assignment groups and; Translated text is in English language.
- As a next step we went on to perform data augmentation in order to take care of imbalance in the input data. One of the benefits of Data augmentation is that it enables one to generate more data from limited data thus helping to address class imbalance issue.
- We kept Group_0 out of augmentation techniques since we had considerable records
- In the initial phase; we used synonym based augmentation technique from ntlk wordnet(word_tokenize); tailoring number of synonyms to be created based on the total records existing in the data frame for each Assignment group. However, we observed that we were not able to pus the accuracy score above 84%. In an effort to improve the model accuracy

we used “nlpaug” package which provides multiple augmentation techniques based on character, words, or sentences.

- We used a combination of word embedding based augmentation, contextual word augmentation method and synonym based augmentation method to create additional records.

Data Augmentation - using https://github.com/makcedward/nlpaug

```
#Install NLPAug Package
!pip install nlpaug
```

```
Collecting nlpaug
  Downloading https://files.pythonhosted.org/packages/03/6d/34f342ba443ca8a74682962f71c465cfcaaa69e9a437cdcf1756986c110d/nlpaug-1.1.2-py3-none-any.whl (387kB)
     |████████████████████████████████| 389kB 11.1MB/s
Installing collected packages: nlpaug
Successfully installed nlpaug-1.1.2
```

```
#Install dependencies for nlpaug
!pip install torch>=1.6.0 transformers>=4.0.0
!pip install nltk>=3.4.5
```

```
#We will use Word Augmentation methods from nlpaug library. nlpaug supports character, word and sentence level augmentation methods
import nlpaug.augmenter.word as naw
```

Extract Glove Embeddings

```
#download the glove embedding zip file from http://nlp.stanford.edu/data/wordvecs/glove.6B.zip
from zipfile import ZipFile
# Check if it is already extracted else Open the zipped file as readonly
if not os.path.isfile('glove.6B/glove.6B.200d.txt'):
    glove_embeddings = 'glove.6B.zip'
    #glove_embeddings = '/content/drive/MyDrive/Capstone/glove.6B.zip'
    with ZipFile(glove_embeddings, 'r') as archive:
        archive.extractall('glove.6B')

# List the files under extracted folder
os.listdir('glove.6B')
```

```
['glove.6B.100d.txt',
 'glove.6B.200d.txt',
 'glove.6B.300d.txt',
 'glove.6B.50d.txt']
```

```
#Word embedding augmentation method
aug1 = naw.WordEmbsAug(model_type='glove', model_path='glove.6B/glove.6B.50d.txt', action="substitute")
aug2 = naw.WordEmbsAug(model_type='glove', model_path='glove.6B/glove.6B.50d.txt', action="insert")
```

```
#Contextual Word augmentation method
aug3 = naw.ContextualWordEmbsAug(model_path='bert-base-uncased', action="substitute")
aug4 = naw.ContextualWordEmbsAug(model_path='roberta-base', action="substitute")
```

- In the above we have extracted Glove embeddings
- Created object to create glove based word embedding augmentation method, and “bert based uncased based” and “Roberta-base” contextual word augmentation method
- Similarly, below synonym based augmentation method using PPDB

```
#Synonym Augmentation method using PPDB models downloaded from http://paraphrase.org/#/download
aug5 = naw.SynonymAug(aug_src='ppdb', model_path='/content/drive/MyDrive/Capstone/ppdb-2.0-s-all')
```

```
#!pip3 install nltk
import nltk
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
from nltk.corpus import wordnet
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
```

```
#Create a new dataframe with records not in GRP_0. We will Augment data for all groups except GRP_0
zero_dataframe = clean_data[clean_data["Assignment group"] == 'GRP_0']
new_dataframe = clean_data[clean_data["Assignment group"] != 'GRP_0']
zero_dataframe.head()
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text
0	spxjnwir pjlcoqds	GRP_0	-verified user details.(employee# & manager na...	-verified user details.(employee and manager n...	en	-verified user details.(employee and manager n...
1	hmjdrvpb komuaywn	GRP_0	\n\nreceived from: hmjdrvpb.komuaywn@gmail.com...	hello team, my meetings/skype meetings etc are...	en	hello team, my meetings/skype meetings etc are...
2	eylqgodm ybqkwiam	GRP_0	\n\nreceived from: eylqgodm.ybqkwiam@gmail.com...	hi cannot log on to vpn best cant log in to vpn	en	hi cannot log on to vpn best cant log in to vpn
3	xbkucsvz gcpydteq	GRP_0	unable to access hr_tool page unable to access...	unable to access hr_tool page unable to access...	en	unable to access hr_tool page unable to access...
4	owlgqjme qhcozdfx	GRP_0	skype error skype error	skype error skype error	no	skype error skype error

```
#Create dataframe copies for different augmentation methods
new_dataframe2 = new_dataframe.copy()
new_dataframe3 = new_dataframe.copy()
new_dataframe4 = new_dataframe.copy()
new_dataframe5 = new_dataframe.copy()
new_dataframe6 = new_dataframe.copy()
```

```
# shape of all dataframes
new_dataframe.shape, zero_dataframe.shape, new_dataframe2.shape, new_dataframe3.shape, new_dataframe4.shape, new_dataframe5.shape
```

```
((4366, 6), (3833, 6), (4366, 6), (4366, 6), (4366, 6), (4366, 6))
```

```
new_dataframe["Augmented_data"] = new_dataframe.apply(lambda x: aug1.augment(x['Translated_Text']),axis=1)
new_dataframe
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text	Augmented_data
6	jyoqwxhz clhxsoqy	GRP_1	event: critical:HostName_221.company.com the v...	event: critical:HostName_.company.com the valu...	en	event: critical:HostName_.company.com the valu...	prix: given: HostName_. company. bbn there val...
17	sigfdwcj reofwzlm	GRP_3	when undocking pc , screen will not come back ...	when undocking pc , screen will not come back ...	en	when undocking pc , screen will not come back ...	when undocking pc, picture will not come home ...
31	kxsceyzo naokumlb	GRP_4	\n\nreceived from: kxsceyzo.naokumlb@gmail.com...	gentles, have two devices that are trying to s...	en	gentles, have two devices that are trying to s...	gentles, that two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	\n\nreceived from: yisohglr.uvteflgb@gmail.com...	hi - the printer printer is not working and ne...	en	hi - the printer printer is not working and ne...	hi - the printer printer gives even working an...
46	bpctwhsn kzqsbmtp	GRP_6	received from: monitoring_tool@company.com\n\n...	job Job_ failed in job_scheduler at: job Job_ ...	en	job Job_ failed in job_scheduler at: job Job_ ...	something Job_ led 2004 job_scheduler at: work...
...	...	...	...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	i am sorry, i have another two accounts that n...	i am sorry, have another two accounts that nee...	en	i am sorry, have another two accounts that nee...	knew am sorry, have another two accounts indee...
8193	cpmaidhj elbaqmtp	GRP_3	tablet needs reimaged due to multiple issues w...	tablet needs reimaged due to multiple issues w...	en	tablet needs reimaged due to multiple issues w...	tablet opportunity reimaged due be usually iss...
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...	good afternoon, am not taken the emails that l...
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...	've anything rest to requires present fitted u...
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...	various prgramdntyme know not be opened a mult...

4366 rows × 7 columns

```
s = new_dataframe.apply(lambda x: pd.Series(x['Augmented_data']), axis=1).stack().reset_index(level=1, drop=True)
s.name = 'Final_Text'
new_dataframe_aug = new_dataframe.drop(['New Description','Augmented_data', 'Clean_Description', 'Translated_Text'],axis=1).join
(s)
new_dataframe_aug
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	prix: given: HostName_. company. bbn there val...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc, picture will not come home ...
31	kxsceyzo naokumlb	GRP_4	en	gentles, that two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer gives even working an...
46	bpctwhsn kzqsbmtp	GRP_6	en	something Job_ led 2004 job_scheduler at: work...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	knew am sorry, have another two accounts indee...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet opportunity reimaged due be usually iss...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, am not taken the emails that l...
8197	ufawcgob aowhxjky	GRP_62	en	've anything rest to requires present fitted u...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme know not be opened a mult...

4366 rows × 4 columns

```
new_dataframe2["Augmented_data"] = new_dataframe2.apply(lambda x: aug2.augment(x['Translated_Text']),axis=1)
new_dataframe2
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text	Augmented_data
6	jyoqwxhz clhxsoqy	GRP_1	event: critical:HostName_221.company.com the v...	event: critical:HostName_.company.com the valu...	en	event: critical:HostName_.company.com the valu...	event: selon critical: frees HostName_. 1828 c...
17	sigfdwcj reofwzlm	GRP_3	when undocking pc , screen will not come back ...	when undocking pc , screen will not come back ...	en	when undocking pc , screen will not come back ...	leavey when cosandey undocking 1597 pc, screen...
31	kxsceyzo naokumlb	GRP_4	\n\nreceived from: kxsceyzo.naokumlb@gmail.com...	gentles, have two devices that are trying to s...	en	gentles, have two devices that are trying to s...	gentles, have two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	\n\nreceived from: yisohglr.uvteflgb@gmail.com...	hi - the printer printer is not working and ne...	en	hi - the printer printer is not working and ne...	mynetworktv-affiliated hi - the printer printe...
46	bpctwhsn kzqsbmtp	GRP_6	received from: monitoring_tool@company.com\n\n...	job Job_ failed in job_scheduler at: job Job_ ...	en	job Job_ failed in job_scheduler at: job Job_ ...	job Job_ failed in job_scheduler at: job Job_ ...
...	...	...	...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	i am sorry, i have another two accounts that n...	i am sorry, have another two accounts that nee...	en	i am sorry, have another two accounts that nee...	i am sorry, have another two accounts that sid...
8193	cpmaidhj elbaqmtp	GRP_3	tablet needs reimaged due to multiple issues w...	tablet needs reimaged due to multiple issues w...	en	tablet needs reimaged due to multiple issues w...	tablet needs reimaged novaes due to multiple v...
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...	good birthdate afternoon, am not 2115 receivin...
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...	recognizance i am unable to sclerites access t...
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...	various prgramdntyme 8.0-9 can tidies not be o...

4366 rows × 7 columns

```
s = new_dataframe2.apply(lambda x: pd.Series(x['Augmented_data']), axis=1).stack().reset_index(level=1, drop=True)
s.name = 'Final_Text'
new_dataframe_aug2 = new_dataframe2.drop(['New Description','Augmented_data', 'Clean_Description', 'Translated_Text'],axis=1).jo
in(s)
new_dataframe_aug2
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	event: selon critical: frees HostName_. 1828 c...
17	sigfdwcj reofwzlm	GRP_3	en	leavey when cosandey undocking 1597 pc, screen...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	mynetworktv-affiliated hi - the printer printe...
46	bpctwhsn kzqsbmtp	GRP_6	en	job Job_ failed in job_scheduler at: job Job_ ...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, have another two accounts that sid...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs reimaged novaes due to multiple v...
8194	avglmrts vhqmtiua	GRP_29	en	good birthdate afternoon, am not 2115 receivin...
8197	ufawcgob aowhxjky	GRP_62	en	recognizance i am unable to sclerites access t...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme 8.0-9 can tidies not be o...

4366 rows × 4 columns

```
new_dataframe3["Augmented_data"] = new_dataframe3.apply(lambda x: aug3.augment(x['Translated_Text']),axis=1)
new_dataframe3
```

```
Token indices sequence length is longer than the specified maximum sequence length for this model (513 > 512). Running this sequence through the model will result in indexing errors
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text	Augmented_data
6	jyoqwxhz clhxsoqy	GRP_1	event: critical:HostName_221.company.com the v...	event: critical:HostName_.company.com the valu...	en	event: critical:HostName_.company.com the valu...	notice : critical : hostname _. systems. com t...
17	sigfdwcj reofwzlm	GRP_3	when undocking pc , screen will not come back ...	when undocking pc , screen will not come back ...	en	when undocking pc , screen will not come back ...	when undocking ended, screen need not came bac...
31	kxsceyzo naokumlb	GRP_4	\n\nreceived from: kxsceyzo.naokumlb@gmail.com...	gentles, have two devices that are trying to s...	en	gentles, have two devices that are trying to s...	gentles, have two printers that are trying to ...
42	yisohglr uvteflgb	GRP_5	\n\nreceived from: yisohglr.uvteflgb@gmail.com...	hi - the printer printer is not working and ne...	en	hi - the printer printer is not working and ne...	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	received from: monitoring_tool@company.com\n\n...	job Job_ failed in job_scheduler at: job Job_ ...	en	job Job_ failed in job_scheduler at: job Job_ ...	job job _ failed against job _ settings at : j...
...	...	...	...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	i am sorry, i have another two accounts that n...	i am sorry, have another two accounts that nee...	en	i am sorry, have another two accounts that nee...	i reply sorry, have another two accounts that ...
8193	cpmaidhj elbaqmtp	GRP_3	tablet needs reimaged due to multiple issues w...	tablet needs reimaged due to multiple issues w...	en	tablet needs reimaged due to multiple issues w...	website needs reimaged due to several factors ...
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...	good afternoon, again not hearing the replies ...
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...	also am unable could access the machine utilit...
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...	these prgramdntyme ports also remain opened in...

4366 rows × 7 columns

```
s = new_dataframe3.apply(lambda x: pd.Series(x['Augmented_data']), axis=1).stack().reset_index(level=1, drop=True)
s.name = 'Final_Text'
new_dataframe_aug3 = new_dataframe3.drop(['New Description','Augmented_data', 'Clean_Description', 'Translated_Text'],axis=1).join(s)
new_dataframe_aug3
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	notice : critical : hostname _. systems. com t...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking ended, screen need not came bac...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have two printers that are trying to ...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	en	job job _ failed against job _ settings at : j...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i reply sorry, have another two accounts that ...
8193	cpmaidhj elbaqmtp	GRP_3	en	website needs reimaged due to several factors ...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, again not hearing the replies ...
8197	ufawcgob aowhxjky	GRP_62	en	also am unable could access the machine utilit...
8198	kqvbrspl jyzoklfx	GRP_49	de	these prgramdntyme ports also remain opened in...

4366 rows × 4 columns

```
new_dataframe4["Augmented_data"] = new_dataframe4.apply(lambda x: aug4.augment(x['Translated_Text']),axis=1)
new_dataframe4
```

```
Token indices sequence length is longer than the specified maximum sequence length for this model (783 > 512). Running this sequence through the model will result in indexing errors
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text	Augmented_
6	jyoqwxhz clhxsoqy	GRP_1	event: critical:HostName_221.company.com the v...	event: critical:HostName_.company.com the valu...	en	event: critical:HostName_.company.com the valu...	ev host:HostName_.company. return val
17	sigfdwcj reofwzlm	GRP_3	when undocking pc , screen will not come back ...	when undocking pc , screen will not come back ...	en	when undocking pc , screen will not come back ...	when undocking pc, here no come down wh
31	kxsceyzo naokumlb	GRP_4	\n\nreceived from: kxsceyzo.naokumlb@gmail.com...	gentles, have two devices that are trying to s...	en	gentles, have two devices that are trying to s...	gentles, have 3 devices are trying to s
42	yisohglr uvteflgb	GRP_5	\n\nreceived from: yisohglr.uvteflgb@gmail.com...	hi - the printer printer is not working and ne...	en	hi - the printer printer is not working and ne...	hi - the printer interface is working ar
46	bpctwhsn kzqsbmtp	GRP_6	received from: monitoring_tool@company.com\n\n...	job Job_ failed in job_scheduler at: job Job_ ...	en	job Job_ failed in job_scheduler at: job Job_ ...	job false_ lock A Object_scheduler at: j
...	...	...	...	...	...	...	
8192	ipwjorsc uboapexr	GRP_10	i am sorry, i have another two accounts that n...	i am sorry, have another two accounts that nee...	en	i am sorry, have another two accounts that nee...	i am sorry, add another files that need
8193	cpmaidhj elbaqmtp	GRP_3	tablet needs reimaged due to multiple issues w...	tablet needs reimaged due to multiple issues w...	en	tablet needs reimaged due to multiple issues w...	tablet needs updated due multiple issue
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...	good afternoon, recommer not receiving any
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...	i am trying not overwrite machine code
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...	various doors can onl opened by shared P

4366 rows × 7 columns

```
s = new_dataframe4.apply(lambda x: pd.Series(x['Augmented_data']), axis=1).stack().reset_index(level=1, drop=True)
s.name = 'Final_Text'
new_dataframe_aug4 = new_dataframe4.drop(['New Description','Augmented_data', 'Clean_Description', 'Translated_Text'],axis=1).join(s)
new_dataframe_aug4
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	event: host:HostName_.company.com return value...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc, here will no come down when...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have 3 devices that are trying to sha...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer interface is not working and ...
46	bpctwhsn kzqsbmtp	GRP_6	en	job false_ lock Again Object_scheduler at: job...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, add another two files that need to...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs updated due any multiple issues i...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, recommended not receiving anym...
8197	ufawcgob aowhxjky	GRP_62	en	i am trying not overwrite the machine code to ...
8198	kqvbrspl jyzoklfx	GRP_49	de	various doors can only be opened by shared PCs...

4366 rows × 4 columns

```
new_dataframe5["Augmented_data"] = new_dataframe5.apply(lambda x: aug5.augment(x['Translated_Text']),axis=1)
new_dataframe5
```

	Caller	Assignment group	New Description	Clean_Description	language	Translated_Text	Augmented_data
6	jyoqwxhz clhxsoqy	GRP_1	event: critical:HostName_221.company.com the v...	event: critical:HostName_.company.com the valu...	en	event: critical:HostName_.company.com the valu...	event: crucial: HostName_. corporation. com th...
17	sigfdwcj reofwzlm	GRP_3	when undocking pc , screen will not come back ...	when undocking pc , screen will not come back ...	en	when undocking pc , screen will not come back ...	when undocking pc, screen will not ceased back...
31	kxsceyzo naokumlb	GRP_4	\n\nreceived from: kxsceyzo.naokumlb@gmail.com...	gentles, have two devices that are trying to s...	en	gentles, have two devices that are trying to s...	gentles, have two devices that address strive ...
42	yisohglr uvteflgb	GRP_5	\n\nreceived from: yisohglr.uvteflgb@gmail.com...	hi - the printer printer is not working and ne...	en	hi - the printer printer is not working and ne...	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	received from: monitoring_tool@company.com\n\n...	job Job_ failed in job_scheduler at: job Job_ ...	en	job Job_ failed in job_scheduler at: job Job_ ...	purposes Job_ accomplished in job_scheduler at...
...	...	...	...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	i am sorry, i have another two accounts that n...	i am sorry, have another two accounts that nee...	en	i am sorry, have another two accounts that nee...	i am sorry, characterized another two accounts...
8193	cpmaidhj elbaqmtp	GRP_3	tablet needs reimaged due to multiple issues w...	tablet needs reimaged due to multiple issues w...	en	tablet needs reimaged due to multiple issues w...	tablet ends reimaged due to multiple subparagr...
8194	avglmrts vhqmtiua	GRP_29	\n\nreceived from: avglmrts.vhqmtiua@gmail.com...	good afternoon, am not receiving the emails th...	en	good afternoon, am not receiving the emails th...	good afternoon, rise not suffering the emails ...
8197	ufawcgob aowhxjky	GRP_62	i am unable to access the machine utilities to...	i am unable to access the machine utilities to...	en	i am unable to access the machine utilities to...	i ben able to access the machinery utilities t...
8198	kqvbrspl jyzoklfx	GRP_49	an mehreren pc`s lassen sich verschiedene prgr...	an mehreren pc`s lassen sich verschiedene prgr...	de	various prgramdntyme can not be opened on mult...	different prgramdntyme can not be re opened on...

4366 rows × 7 columns

```
s = new_dataframe5.apply(lambda x: pd.Series(x['Augmented_data']), axis=1).stack().reset_index(level=1, drop=True)
s.name = 'Final_Text'
new_dataframe_aug5 = new_dataframe5.drop(['New Description','Augmented_data', 'Clean_Description', 'Translated_Text'],axis=1).join(s)
new_dataframe_aug5
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	event: crucial: HostName_. corporation. com th...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc, screen will not ceased back...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have two devices that address strive ...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	en	purposes Job_ accomplished in job_scheduler at...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, characterized another two accounts...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet ends reimaged due to multiple subparagr...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, rise not suffering the emails ...
8197	ufawcgob aowhxjky	GRP_62	en	i ben able to access the machinery utilities t...
8198	kqvbrspl jyzoklfx	GRP_49	de	different prgramdntyme can not be re opened on...

4366 rows × 4 columns

```
# Grp_0 dataframe
zero_dataframe = zero_dataframe.rename(columns={"Translated_Text": "Final_Text"})
zero_dataframe = zero_dataframe.drop(['New Description', 'Clean_Description'], axis = 1)
```

```
zero_dataframe
```

	Caller	Assignment group	language	Final_Text
0	spxjnwir pjlcoqds	GRP_0	en	-verified user details.(employee and manager n...
1	hmjdrvpb komuaywn	GRP_0	en	hello team, my meetings/skype meetings etc are...
2	eylqgodm ybqkwiam	GRP_0	en	hi cannot log on to vpn best cant log in to vpn
3	xbkucsvz gcpydteq	GRP_0	en	unable to access hr_tool page unable to access...
4	owlgqjme qhcozdfx	GRP_0	no	skype error skype error
...	...	...	...	...
8187	rbozivdq gmlhrtvp	GRP_0	en	name:mfeyouli ndobtzpw language: browser:micro...
8188	sdvlxbfe ptnahjkw	GRP_0	en	account locked account locked
8191	tmopbken ibzougsd	GRP_0	en	hr_tool etime option not visitble hr_tool etim...
8195	rbozivdq gmlhrtvp	GRP_0	en	telephony_software issue telephony_software issue
8196	oybwdsgx oxyhwrfz	GRP_0	en	vip: windows password reset for tifpdchb pedxr...

3833 rows × 4 columns

```
#Original data (without Augmentation)

new_dataframe6 = new_dataframe6.rename(columns={"Translated_Text": "Final_Text"})
new_dataframe6 = new_dataframe6.drop(['New Description', 'Clean_Description'], axis = 1)
new_dataframe6
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	event: critical:HostName_.company.com the valu...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc , screen will not come back ...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	en	job Job_ failed in job_scheduler at: job Job_ ...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, have another two accounts that nee...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs reimaged due to multiple issues w...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, am not receiving the emails th...
8197	ufawcgob aowhxjky	GRP_62	en	i am unable to access the machine utilities to...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme can not be opened on mult...

4366 rows × 4 columns

```
new_dataframe6
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	event: critical:HostName_.company.com the valu...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc , screen will not come back ...
31	kxsceyzo naokumlb	GRP_4	en	gentles, have two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer is not working and ne...
46	bpctwhsn kzqsbmtp	GRP_6	en	job Job_ failed in job_scheduler at: job Job_ ...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, have another two accounts that nee...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs reimaged due to multiple issues w...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, am not receiving the emails th...
8197	ufawcgob aowhxjky	GRP_62	en	i am unable to access the machine utilities to...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme can not be opened on mult...

4366 rows × 4 columns

```
# Concat all the augmented frames
dataframes=[new_dataframe_aug, new_dataframe_aug2, new_dataframe_aug3, new_dataframe_aug4, new_dataframe_aug5, zero_dataframe, n
ew_dataframe6]
clean_data_result= pd.concat(dataframes)
clean_data_result
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	prix: given: HostName_. company. bbn there val...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc, picture will not come home ...
31	kxsceyzo naokumlb	GRP_4	en	gentles, that two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer gives even working an...
46	bpctwhsn kzqsbmtp	GRP_6	en	something Job_ led 2004 job_scheduler at: work...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, have another two accounts that nee...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs reimaged due to multiple issues w...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, am not receiving the emails th...
8197	ufawcgob aowhxjky	GRP_62	en	i am unable to access the machine utilities to...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme can not be opened on mult...

30029 rows × 4 columns

```
#Remove any duplicate rows after augmentation
clean_data_result = clean_data_result.drop_duplicates(subset='Final_Text', keep="first")
```

```
# Serialize the Augmented dataset for later use
clean_data_result.to_csv('Interim_data.csv', index=False, encoding='utf_8_sig')
#with open('/content/Interim_data.pkl','wb') as f:
with open('/content/drive/MyDrive/Capstone/Interim_data.pkl','wb') as f:
    pickle.dump(clean_data_result, f, pickle.HIGHEST_PROTOCOL)
```

Interim_data.pkl file will be used later for DL models as we wont be performing stop words removal and lemmatisation on it.

```
# Load the consolidated final translated pickle file
#with open('/content/drive/MyDrive/Capstone/Interim_data.pkl','rb') as f:
with open('/content/drive/MyDrive/Capstone/Interim_data.pkl','rb') as f:
    clean_data_result = pickle.load(f)
```

```
clean_data_result
```

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	prix: given: HostName_. company. bbn there val...
17	sigfdwcj reofwzlm	GRP_3	en	when undocking pc, picture will not come home ...
31	kxsceyzo naokumlb	GRP_4	en	gentles, that two devices that are trying to s...
42	yisohglr uvteflgb	GRP_5	en	hi - the printer printer gives even working an...
46	bpctwhsn kzqsbmtp	GRP_6	en	something Job_ led 2004 job_scheduler at: work...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	i am sorry, have another two accounts that nee...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet needs reimaged due to multiple issues w...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon, am not receiving the emails th...
8197	ufawcgob aowhxjky	GRP_62	en	i am unable to access the machine utilities to...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme can not be opened on mult...

28386 rows × 4 columns

greatlearning
Learning for Life

Stop words removal and Lemmatise text

In [87]:
```
clean_data_result.isnull().sum()
```

Out[87]:
```
Caller             0
Assignment group   0
language           0
Final_Text         0
dtype: int64
```

In [88]:
```
clean_data_result['Final_Text'] = clean_data_result['Final_Text'].fillna("")
```

In [89]:
```
import re
import string
nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer

stop_words = set(stopwords.words('english'))

processed_all_documents = list()

for desc in clean_data_result['Final_Text']:
    word_tokens = word_tokenize(desc)

    filtered_sentence = []

    # Removing Stopwords
    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(w)

    words = ' '.join(filtered_sentence)
    processed_all_documents.append(words)
```

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

In [90]:
```
clean_data_result['Final_Text'] = processed_all_documents
```

In [91]:
```
clean_data_result.head(50)
```

Out[91]:

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	prix : given : HostName_ . company . bbn value...
17	sigfdwcj reofwzlm	GRP_3	en	undocking pc , picture come home 8:02 pc , scr...
31	kxsceyzo naokumlb	GRP_4	en	gentles , two devices trying share ip address ...
42	yisohglr uvteflgb	GRP_5	en	hi - printer printer gives even working needs ...
46	bpctwhsn kzqsbmtp	GRP_6	en	something Job_ led 2004 job_scheduler : workin...
48	aofnvyzt eqiyskhm	GRP_7	en	away closing call , making agent keeps , `` ac...
49	bpctwhsn kzqsbmtp	GRP_8	en	go mm_zscr_dly_merktc turn job_scheduler : job...
56	wckrxovs aunsgzmd	GRP_6	en	dn came material / plant plant_ / pcs without ...
58	bpctwhsn kzqsbmtp	GRP_8	en	job mm_zscr_dly_merktc failed job_scheduler ho...
59	bpctwhsn kzqsbmtp	GRP_8	en	come Job_ although job_scheduler : Job_ failed...
62	qfnthlam lxvnwuja	GRP_6	en	please leave , try create dlv rather mm ' espí...
63	utyeofsk rdyzpwhi	GRP_8	en	apac sales : 11 switches much quickly since le...
66	bpctwhsn kzqsbmtp	GRP_9	en	job Job_ failed job_scheduler : jobs Job_ fail...
67	bpctwhsn kzqsbmtp	GRP_8	en	keeping Job_ help job_scheduler : job Job_ job...
68	bpctwhsn kzqsbmtp	GRP_8	en	job Job_ failed job_scheduler university : tro...
69	bpctwhsn kzqsbmtp	GRP_8	en	find Job_ move job_scheduler morning : job Job...
74	bpctwhsn kzqsbmtp	GRP_10	en	job hr_payroll_na_u giving job_scheduler : wel...
75	bpctwhsn kzqsbmtp	GRP_10	en	job hr_payroll_na_u failed job_scheduler last ...
76	bpctwhsn kzqsbmtp	GRP_10	en	job hr_payroll_na_u failed ago job_scheduler c...
77	mnlazfsr mtqrkhnx	GRP_8	en	type outage : _____network _____circuit ___x__...
78	jyoqwxhz clhxsoqy	GRP_8	en	type outage : ___x__network _____circuit _____...
79	bpctwhsn kzqsbmtp	GRP_6	en	got Job_ job_scheduler : perhaps Job_ taking j...
81	bpctwhsn kzqsbmtp	GRP_9	en	time bwhrattr failed job_scheduler leaving : c...
82	vlymsnej whlqxcst	GRP_11	en	hello providing , need monitor manufacturing d...
83	bpctwhsn kzqsbmtp	GRP_6	en	apo_bop_plant_a quickly following job_schedule...
84	bpctwhsn kzqsbmtp	GRP_9	en	Job_ failed job_scheduler : job Job_ failed se...
85	bpctwhsn kzqsbmtp	GRP_8	en	job mm_zscr_wkly_rollfgyuej job_scheduler time...
87	bpctwhsn kzqsbmtp	GRP_8	en	job Job_ failed place job_scheduler : talking ...
88	bpctwhsn kzqsbmtp	GRP_5	en	job SID_cold failed opened job_scheduler : job...
89	bpctwhsn kzqsbmtp	GRP_5	en	rest SID_cold successfully job_scheduler : hir...
90	bpctwhsn kzqsbmtp	GRP_5	en	job Job_ finally started job_scheduler : job J...
91	bpctwhsn kzqsbmtp	GRP_5	en	're SID_cold move job_scheduler : job SID_cold...
92	bpctwhsn kzqsbmtp	GRP_5	en	Job_ came job_scheduler : job Job_ failed firs...
93	bpctwhsn kzqsbmtp	GRP_5	en	job SID_cold failed also job_scheduler : job S...
94	bpctwhsn kzqsbmtp	GRP_5	en	job SID_cold successfully job_scheduler friday...
95	jyoqwxhz clhxsoqy	GRP_12	en	amssm : h : \ remixed : sys - amssm ef compati...
100	bpctwhsn kzqsbmtp	GRP_9	en	job Job_ failed job_scheduler : job Job_ final...
101	bpctwhsn kzqsbmtp	GRP_8	en	though Job_ unable job_scheduler : job Job_ fa...
102	bpctwhsn kzqsbmtp	GRP_6	en	job Job_ failed job_scheduler moved : ever Job...

```
#Lemmatisation using spacy library
!pip3 install spacy
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.6/dist-packages (2.2.4)
```

In [94]:
```
import spacy

nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])
allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV']
def lemmatize_text(text):
    doc = nlp(text)
    return ' '.join([token.lemma_ for token in doc])

clean_data_result['Final_Text'] = clean_data_result['Final_Text'].apply(lemmatize_text)
```

In [95]:
```
clean_data_result
```

Out[95]:

	Caller	Assignment group	language	Final_Text
6	jyoqwxhz clhxsoqy	GRP_1	en	prix : give : HostName _ . company . bbn value...
17	sigfdwcj reofwzlm	GRP_3	en	undock pc , picture come home 8:02 pc , screen...
31	kxsceyzo naokumlb	GRP_4	en	gentle , two device try share ip address . try...
42	yisohglr uvteflgb	GRP_5	en	hi - printer printer give even work need part ...
46	bpctwhsn kzqsbmtp	GRP_6	en	something Job _ lead 2004 job_scheduler : work...
...	...	...	...	...
8192	ipwjorsc uboapexr	GRP_10	en	sorry , another two account need add : . pleas...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet need reimage due multiple issue crm wif...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon , receive email send zz mail . ...
8197	ufawcgob aowhxjky	GRP_62	en	unable access machine utility finish drawer ad...
8198	kqvbrspl jyzoklfx	GRP_49	de	various prgramdntyme open multiple pc . cnc ar...

28386 rows × 4 columns

In [96]:
```
# Serialize the translated dataset
clean_data_result.to_csv('Final_data.csv', index=False, encoding='utf_8_sig')
with open('/content/Final_data.pkl','wb') as f:
    pickle.dump(clean_data_result, f, pickle.HIGHEST_PROTOCOL)
```

In [97]:
```
# Load the translated pickle file
with open('/content/Final_data.pkl','rb') as f:
    clean_data = pickle.load(f)
```

Manual inspection showed that groups having Monitoring tool emails and Job scheduler errors are dropping the accuracy as they dont have paterns for Machine Learning algorithms.

Moving them into 1 seperate group called "GRP_MonitoringTool".

```
#Moving all the records coming from monitoring tool/job scheduler to one group

clean_data.loc[(clean_data['Final_Text'].astype(str).str.contains('scheduler')), 'Assignment group'] = "GRP_MonitoringTool"
```

```
assignment_group_cnt=clean_data['Assignment group'].value_counts()
assignment_group_cnt.describe()
```

```
count      63.000000
mean      450.571429
std       769.162526
min         6.000000
25%        67.500000
50%       180.000000
75%       494.000000
max      4749.000000
Name: Assignment group, dtype: float64
```

```
assignment_group_cnt = clean_data['Assignment group'].value_counts().rename_axis('Grp_name').reset_index(name='counts')
```

```
assignment_group_cnt
```

	Grp_name	counts
0	GRP_MonitoringTool	4749
1	GRP_0	3188
2	GRP_24	1681
3	GRP_8	1518
4	GRP_12	1419
...	...	...
58	GRP_72	11
59	GRP_66	6
60	GRP_68	6
61	GRP_55	6
62	GRP_57	6

63 rows × 2 columns

greatlearning
Learning for Life

Also noticeds that groups having less than 500 records post augmentation is dropping overall model efficiency.

Resampling the groups having less than 500 records to take care of above concern

```
#Sory for the messy code, didnt have patience to write a lambda function. Splitting dataframes with < 500 records for resampling
temp_dataframe = clean_data[(clean_data["Assignment group"] == 'GRP_43') | (clean_data["Assignment group"] == 'GRP_46') | (clean
_data["Assignment group"] == 'GRP_59') | (clean_data["Assignment group"] == 'GRP_49') | (clean_data["Assignment group"] == 'GRP_
60') | (clean_data["Assignment group"] == 'GRP_51') | (clean_data["Assignment group"] == 'GRP_52') | (clean_data["Assignment gro
up"] == 'GRP_65') | (clean_data["Assignment group"] == 'GRP_53') | (clean_data["Assignment group"] == 'GRP_39') | (clean_data["A
ssignment group"] == 'GRP_36') | (clean_data["Assignment group"] == 'GRP_50') | (clean_data["Assignment group"] == 'GRP_44') |
(clean_data["Assignment group"] == 'GRP_47') | (clean_data["Assignment group"] == 'GRP_37') | (clean_data["Assignment group"] ==
'GRP_27') | (clean_data["Assignment group"] == 'GRP_5') | (clean_data["Assignment group"] == 'GRP_1') | (clean_data["Assignment
group"] == 'GRP_62') | (clean_data["Assignment group"] == 'GRP_23') | (clean_data["Assignment group"] == 'GRP_17') | (clean_data
["Assignment group"] == 'GRP_48') | (clean_data["Assignment group"] == 'GRP_45') | (clean_data["Assignment group"] == 'GRP_21')
| (clean_data["Assignment group"] == 'GRP_11') | (clean_data["Assignment group"] == 'GRP_22') | (clean_data["Assignment group"]
== 'GRP_20') | (clean_data["Assignment group"] == 'GRP_42') | (clean_data["Assignment group"] == 'GRP_30') | (clean_data["Assign
ment group"] == 'GRP_15') | (clean_data["Assignment group"] == 'GRP_41') | (clean_data["Assignment group"] == 'GRP_28') | (clean
_data["Assignment group"] == 'GRP_40') | (clean_data["Assignment group"] == 'GRP_26') | (clean_data["Assignment group"] == 'GRP_
34') | (clean_data["Assignment group"] == 'GRP_6') | (clean_data["Assignment group"] == 'GRP_7') | (clean_data["Assignment grou
p"] == 'GRP_31') | (clean_data["Assignment group"] == 'GRP_9') | (clean_data["Assignment group"] == 'GRP_10')]
temp_dataframe2 = clean_data[(clean_data["Assignment group"] != 'GRP_43') & (clean_data["Assignment group"] != 'GRP_46') & (clea
n_data["Assignment group"] != 'GRP_59') & (clean_data["Assignment group"] != 'GRP_49') & (clean_data["Assignment group"] != 'GRP
_60') & (clean_data["Assignment group"] != 'GRP_51') & (clean_data["Assignment group"] != 'GRP_52') & (clean_data["Assignment gr
oup"] != 'GRP_65') & (clean_data["Assignment group"] != 'GRP_53') & (clean_data["Assignment group"] != 'GRP_39') & (clean_data
["Assignment group"] != 'GRP_36') & (clean_data["Assignment group"] != 'GRP_50') & (clean_data["Assignment group"] != 'GRP_44')
& (clean_data["Assignment group"] != 'GRP_47') & (clean_data["Assignment group"] != 'GRP_37') & (clean_data["Assignment group"]
!= 'GRP_27') & (clean_data["Assignment group"] != 'GRP_5') & (clean_data["Assignment group"] != 'GRP_1') & (clean_data["Assignme
nt group"] != 'GRP_62') & (clean_data["Assignment group"] != 'GRP_23') & (clean_data["Assignment group"] != 'GRP_17') & (clean_d
ata["Assignment group"] != 'GRP_48') & (clean_data["Assignment group"] != 'GRP_45') & (clean_data["Assignment group"] != 'GRP_2
1') & (clean_data["Assignment group"] != 'GRP_11') & (clean_data["Assignment group"] != 'GRP_22') & (clean_data["Assignment grou
p"] != 'GRP_20') & (clean_data["Assignment group"] != 'GRP_42') & (clean_data["Assignment group"] != 'GRP_30') & (clean_data["As
signment group"] != 'GRP_15') & (clean_data["Assignment group"] != 'GRP_41') & (clean_data["Assignment group"] != 'GRP_28') & (c
lean_data["Assignment group"] != 'GRP_40') & (clean_data["Assignment group"] != 'GRP_26') & (clean_data["Assignment group"] !=
'GRP_34') & (clean_data["Assignment group"] != 'GRP_6') & (clean_data["Assignment group"] != 'GRP_7') & (clean_data["Assignment
group"] != 'GRP_31') & (clean_data["Assignment group"] != 'GRP_9') & (clean_data["Assignment group"] != 'GRP_10')]
temp_dataframe2
```

	Caller	Assignment group	language	Final_Text
17	sigfdwcj reofwzlm	GRP_3	en	undock pc , picture come home 8:02 pc , screen...
31	kxsceyzo naokumlb	GRP_4	en	gentle , two device try share ip address . try...
46	bpctwhsn kzqsbmtp	GRP_MonitoringTool	en	something Job _ lead 2004 job_scheduler : work...
49	bpctwhsn kzqsbmtp	GRP_MonitoringTool	en	go mm_zscr_dly_merktc turn job_scheduler : job...
58	bpctwhsn kzqsbmtp	GRP_MonitoringTool	en	job mm_zscr_dly_merktc fail job_scheduler hour...
...	...	...	...	...
8183	hugcadrn ixhlwdgt	GRP_2	en	please remove user hugcadrn ixhlwdgt (ralftei...
8186	pvbomqht smfkuhwi	GRP_3	en	pc receive multiple window security update ear...
8189	mpihysnw wrctgoan	GRP_29	en	please contact ed pasgryowski (pasgryo) purc...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet need reimage due multiple issue crm wif...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon , receive email send zz mail . ...

21325 rows × 4 columns

```
assignment_group_cnt2 = temp_dataframe2['Assignment group'].value_counts().rename_axis('Grp_name').reset_index(name='counts')
```

```
assignment_group_cnt2
```

	Grp_name	counts
0	GRP_MonitoringTool	4749
1	GRP_0	3188
2	GRP_24	1681
3	GRP_8	1518
4	GRP_12	1419
5	GRP_19	1281
6	GRP_3	1197
7	GRP_2	1191
8	GRP_13	847
9	GRP_14	688
10	GRP_25	671
11	GRP_33	640
12	GRP_4	593
13	GRP_29	565
14	GRP_18	510
15	GRP_16	510
16	GRP_32	24
17	GRP_56	18
18	GRP_72	11
19	GRP_68	6
20	GRP_57	6
21	GRP_66	6
22	GRP_55	6

greatlearning
Learning for Life

```
#resampling all groups having less than 500 records
from sklearn.utils import resample
clean_data_resampled = temp_dataframe[0:0]
for grp in temp_dataframe['Assignment group'].unique():
    temp_dataframe1 = temp_dataframe.apply(lambda x : True
            if str(temp_dataframe['Assignment group']) == grp else False, axis = 1)
    num_rows = len(temp_dataframe1[temp_dataframe1 == True].index)
    if(num_rows < 500):
            temp_dataframeGrpDF = temp_dataframe[temp_dataframe['Assignment group'] == grp]
            resampled = resample(temp_dataframeGrpDF, replace=True, n_samples=500, random_state=123)
            clean_data_resampled = clean_data_resampled.append(resampled)
```

```
clean_data_resampled
```

	Caller	Assignment group	language	Final_Text
2281	hwbukcsm hwobikcv	GRP_1	en	hi , try fluctuate password poruxnwb yfaqhceo ...
1306	xagyhbio jvrdnpkh	GRP_1	fr	job get fail due dbif_rsql_sql_error issue job...
4454	mnlazfsr mtqrkhnx	GRP_1	en	also shop _ floor _ app host : unreachable con...
6	jyoqwxhz clhxsoqy	GRP_1	en	event : crucial : HostName _ . corporation . c...
4454	mnlazfsr mtqrkhnx	GRP_1	en	call shop_floor_app reporting : penetrable sta...
...	...	...	...	...
5527	htnvbwxs gwfrzuex	GRP_65	en	good nmfs day . please assist : please rabonza...
3696	urvitans laqdwvgo	GRP_65	en	frese user delete deshaie update , run psychog...
4808	urvitans laqdwvgo	GRP_65	en	seem continued com update kb get download user...
6857	koahsriq wdugqatr	GRP_65	en	good < unk > need rebuild win laptop new win i...
6314	hkrecpfv kgwpbexv	GRP_65	en	hello . n't believe computer download new syma...

20000 rows × 4 columns

In [107]:
```
#Concat dataframes post resampling
dataframes=[clean_data_resampled, temp_dataframe2]
clean_data_resampled= pd.concat(dataframes)
clean_data_resampled
```

Out[107]:

	Caller	Assignment group	language	Final_Text
2281	hwbukcsm hwobikcv	GRP_1	en	hi , try fluctuate password poruxnwb yfaqhceo ...
1306	xagyhbio jvrdnpkh	GRP_1	fr	job get fail due dbif_rsql_sql_error issue job...
4454	mnlazfsr mtqrkhnx	GRP_1	en	also shop _ floor _ app host : unreachable con...
6	jyoqwxhz clhxsoqy	GRP_1	en	event : crucial : HostName _ . corporation . c...
4454	mnlazfsr mtqrkhnx	GRP_1	en	call shop_floor_app reporting : penetrable sta...
...	...	...	...	...
8183	hugcadrn ixhlwdgt	GRP_2	en	please remove user hugcadrn ixhlwdgt (ralftei...
8186	pvbomqht smfkuhwi	GRP_3	en	pc receive multiple window security update ear...
8189	mpihysnw wrctgoan	GRP_29	en	please contact ed pasgryowski (pasgryo) purc...
8193	cpmaidhj elbaqmtp	GRP_3	en	tablet need reimage due multiple issue crm wif...
8194	avglmrts vhqmtiua	GRP_29	en	good afternoon , receive email send zz mail . ...

41325 rows × 4 columns

In [108]:
```
assignment_group_cnt3 = clean_data_resampled['Assignment group'].value_counts().rename_axis('Grp_name').reset_index(name='counts')
assignment_group_cnt3
```

Out[108]:

	Grp_name	counts
0	GRP_MonitoringTool	4749
1	GRP_0	3188
2	GRP_24	1681
3	GRP_8	1518
4	GRP_12	1419
...	...	...
58	GRP_72	11
59	GRP_55	6
60	GRP_57	6
61	GRP_68	6
62	GRP_66	6

63 rows × 2 columns

In [109]:
```
# Saving Final pkl file after resampling
clean_data_resampled.to_csv('Final_data_resampled.csv', index=False, encoding='utf_8_sig')
#with open('/content/Final_data.pkl','wb') as f:
with open('Final_data_resampled.pkl','wb') as f:
    pickle.dump(clean_data_resampled, f, pickle.HIGHEST_PROTOCOL)
```

Load from here to save time while model training.

:
```
# Load the translated pickle file. Load from here to save time.
#with open('/content/drive/MyDrive/Capstone/Final_data_resampled.pkl','rb') as f:
with open('Final_data_resampled.pkl','rb') as f:
    clean_data = pickle.load(f)
```

Univariate visualization

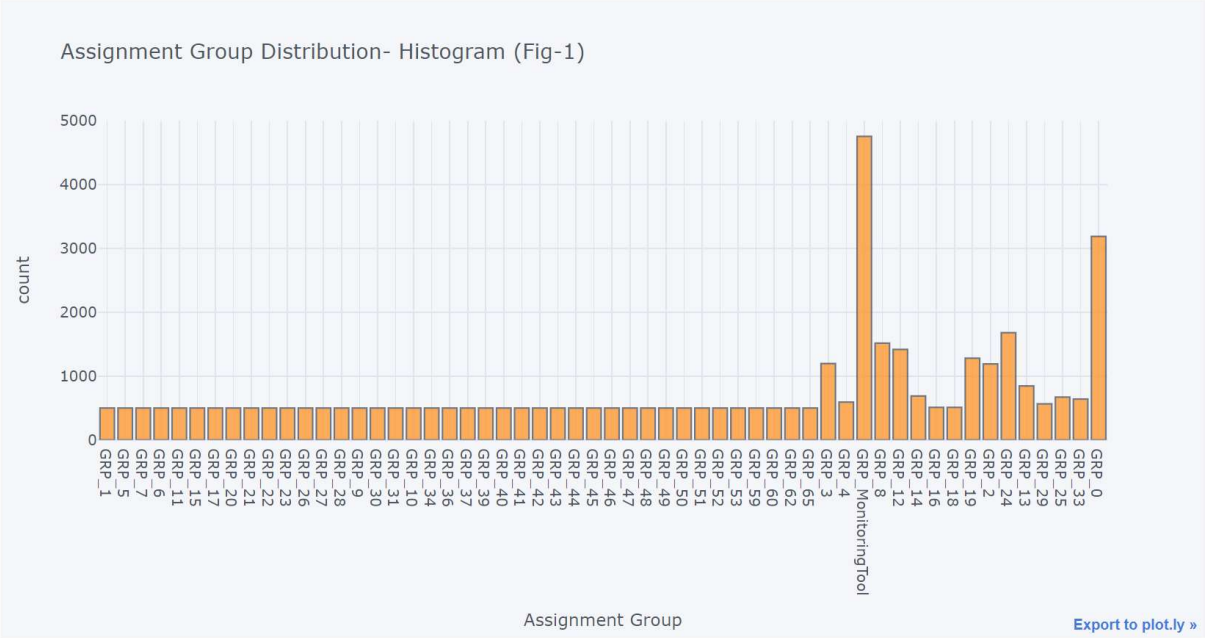
Single-variable or univariate visualization is the simplest type of visualization which consists of observations on only a single characteristic or attribute. Univariate visualization includes histogram, bar plots and line charts.

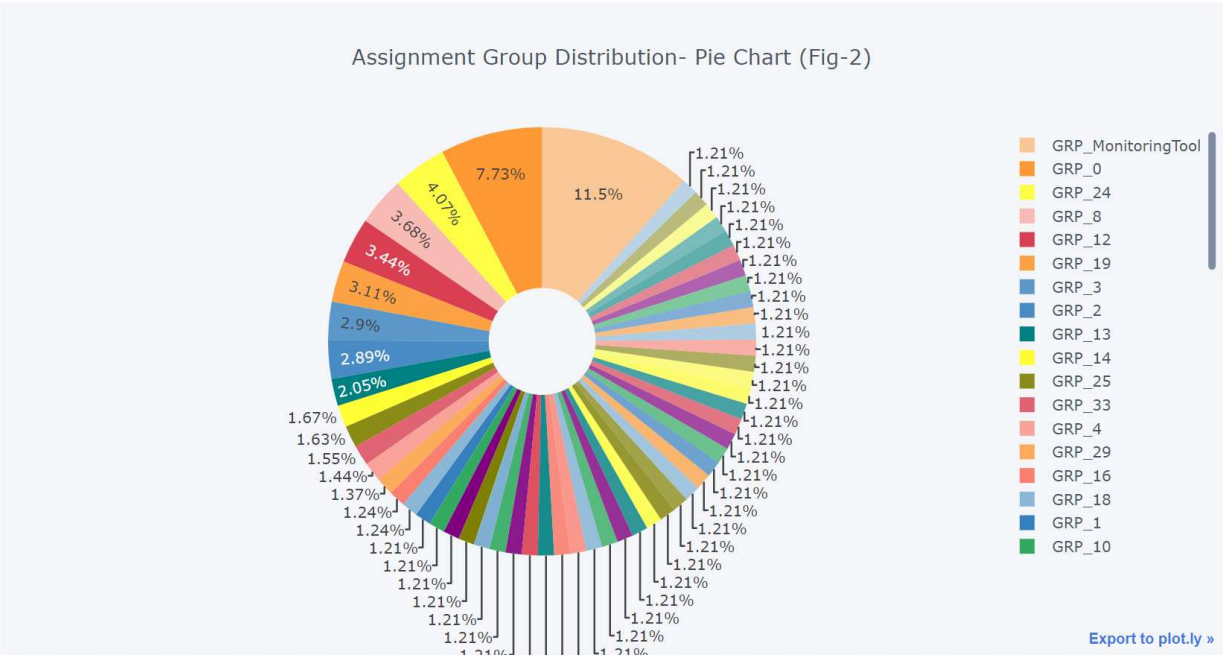
The distribution of Assignment groups

Plots how the assignments groups are scattered across the dataset. The bar chart, histogram and pie chart tells the frequency of any ticket assigned to any group OR the tickets count for each group.

```
# Assignment group distribution
print('\033[1mTotal assignment groups:\033[0m', clean_data['Assignment group'].nunique())

# Histogram
clean_data['Assignment group'].iplot(
    kind='hist',
    xTitle='Assignment Group',
    yTitle='count',
    title='Assignment Group Distribution- Histogram (Fig-1)')

# Pie chart
assgn_grp = pd.DataFrame(clean_data.groupby('Assignment group').size(),columns = ['Count']).reset_index()
assgn_grp.iplot(
    kind='pie',
    labels='Assignment group',
    values='Count',
    title='Assignment Group Distribution- Pie Chart (Fig-2)',
    hoverinfo="label+percent+name", hole=0.25)
```

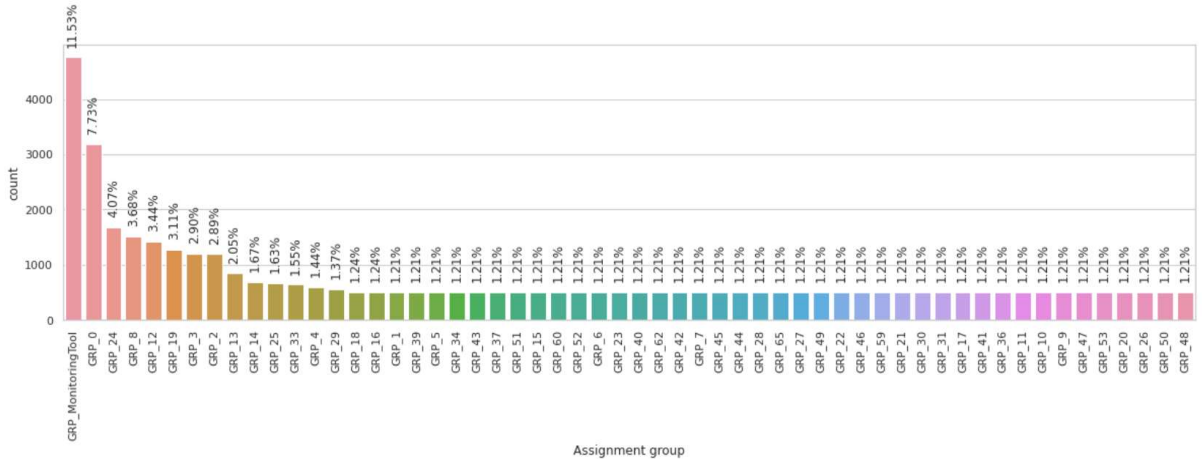
Total assignment groups: 56





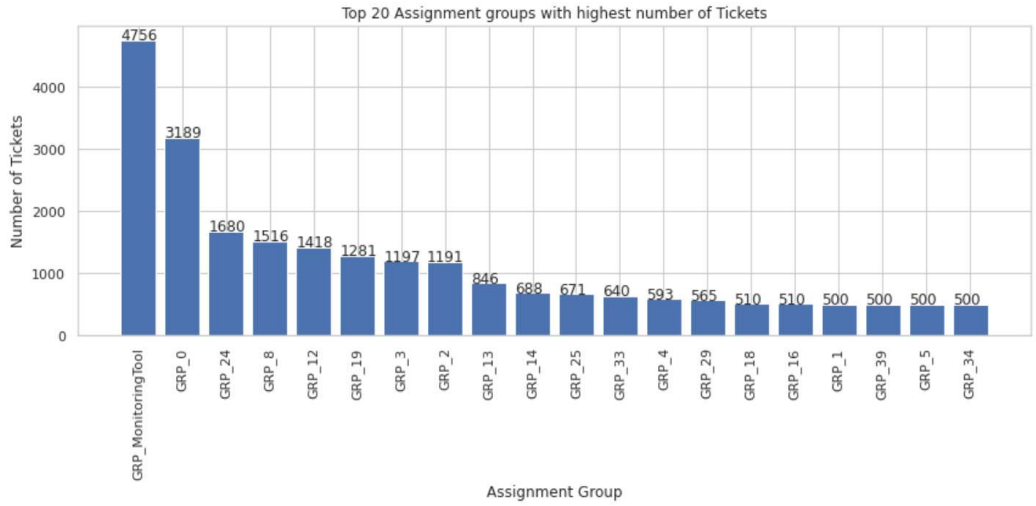
Lets visualize the percentage of incidents per assignment group

```
# Plot to visualize the percentage data distribution across different groups
sns.set(style="whitegrid")
plt.figure(figsize=(20,5))
ax = sns.countplot(x="Assignment group", data=clean_data, order=clean_data["Assignment group"].value_counts().index)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
for p in ax.patches:
  ax.annotate(str(format(p.get_height()/len(clean_data.index)*100, '.2f')+"%"), (p.get_x() + p.get_width() / 2., p.get_height
()), ha = 'center', va = 'bottom', rotation=90, xytext = (0, 10), textcoords = 'offset points')
```



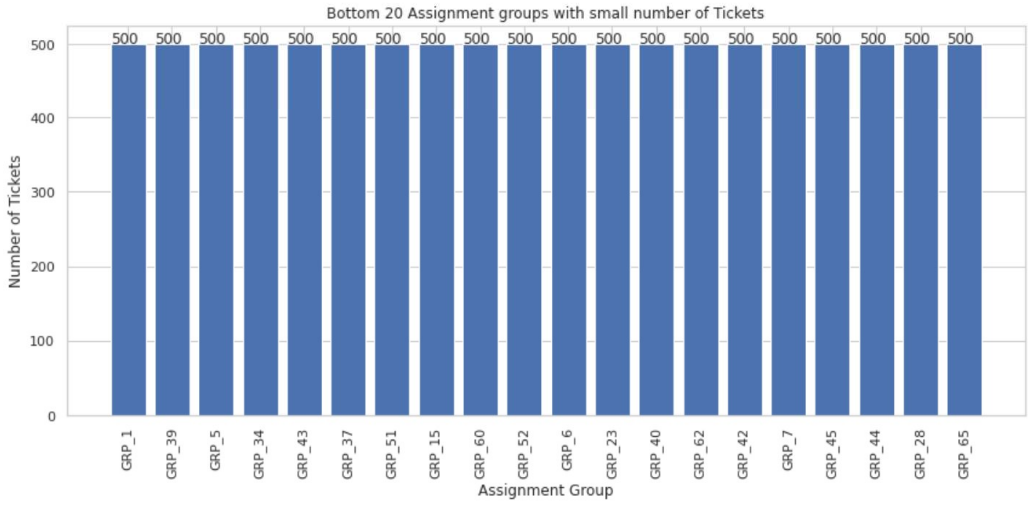
```
top_20 = clean_data['Assignment group'].value_counts().nlargest(20).reset_index()
```

```
plt.figure(figsize=(12,6))
bars = plt.bar(top_20['index'],top_20['Assignment group'])
plt.title('Top 20 Assignment groups with highest number of Tickets')
plt.xlabel('Assignment Group')
plt.xticks(rotation=90)
plt.ylabel('Number of Tickets')

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x(), yval + .005, yval)
plt.tight_layout()
plt.show()
```



```
bottom_20 = clean_data['Assignment group'].value_counts().nsmallest(20).reset_index()
```

```
plt.figure(figsize=(12,6))
bars = plt.bar(bottom_20['index'],bottom_20['Assignment group'])
plt.title('Bottom 20 Assignment groups with small number of Tickets')
plt.xlabel('Assignment Group')
plt.xticks(rotation=90)
plt.ylabel('Number of Tickets')
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x(), yval + .005, yval)
plt.tight_layout()
plt.show()
```

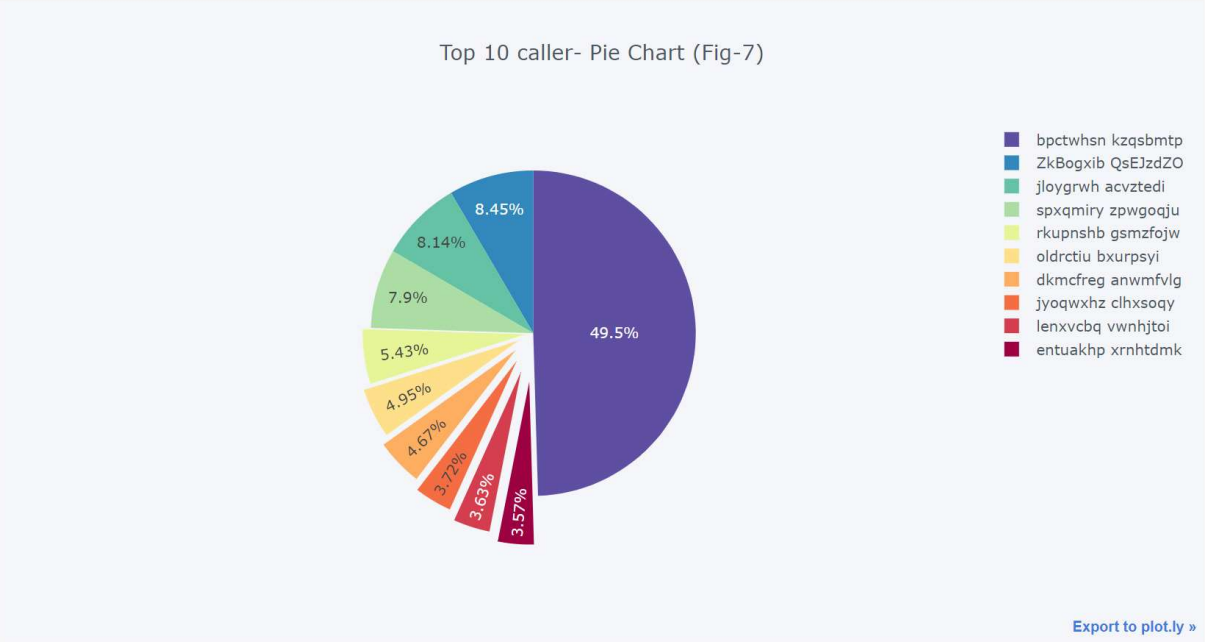


The distribution of Callers

Plots how the callers are associated with tickets and what are the assignment groups they most frequently raise tickets for.

```
# Find out top 10 callers in terms of frequency of raising tickets in the entire dataset
print('\033[1mTotal caller count:\033[0m', clean_data['Caller'].nunique())
df = pd.DataFrame(clean_data.groupby(['Caller']).size().nlargest(10), columns=['Count']).reset_index()
df.iplot(kind='pie',
         labels='Caller',
         values='Count',
         title='Top 10 caller- Pie Chart (Fig-7)',
         colorscale='-spectral',
         pull=[0,0,0,0,0.05,0.1,0.15,0.2,0.25,0.3])
```

Total caller count: 2676



```
# Top 5 callers in each assignment group
top_n = 5
s = clean_data['Caller'].groupby(clean_data['Assignment group']).value_counts()
caller_grp = pd.DataFrame(s.groupby(level=0).nlargest(top_n).reset_index(level=0, drop=True))
caller_grp.head(15)
```

		Caller
Assignment group	**Caller**	
GRP_0	**rbozivdq gmlhrtvp**	60
	efbwiadp dicafxhv	44
	olckhmvx pcqobjnd	29
	fumkcsji sarmtlhy	26
	mfeyouli ndobtzpw	13
GRP_1	**jloygrwh acvztedi**	86
	spxqmiry zpwgoqju	60
	mnlazfsr mtqrkhnx	57
	jyoqwxhz clhxsoqy	56
	kbnfxpsy gehxzayq	48
GRP_10	**ihfkwzjd erbxoyqk**	42
	dizquolf hlykecxa	36
	ipwjorsc uboapexr	19
	miecoszw mhvbnodw	18
	lokiwfhg udkoqrcg	17

The distribution of description lengths

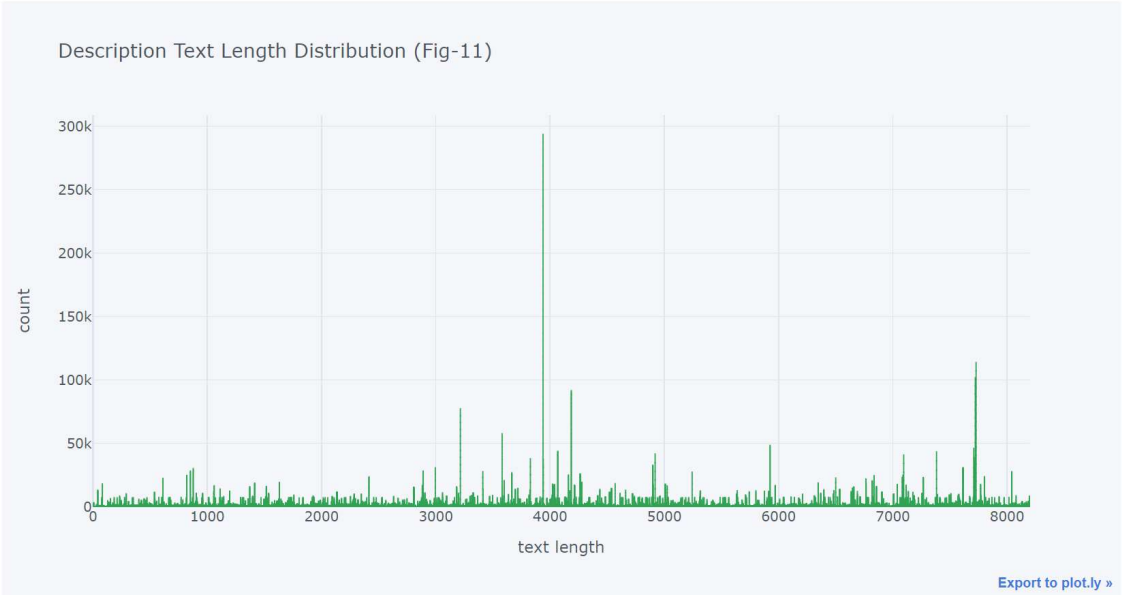
Plots the variation of length and word count of new description attribute

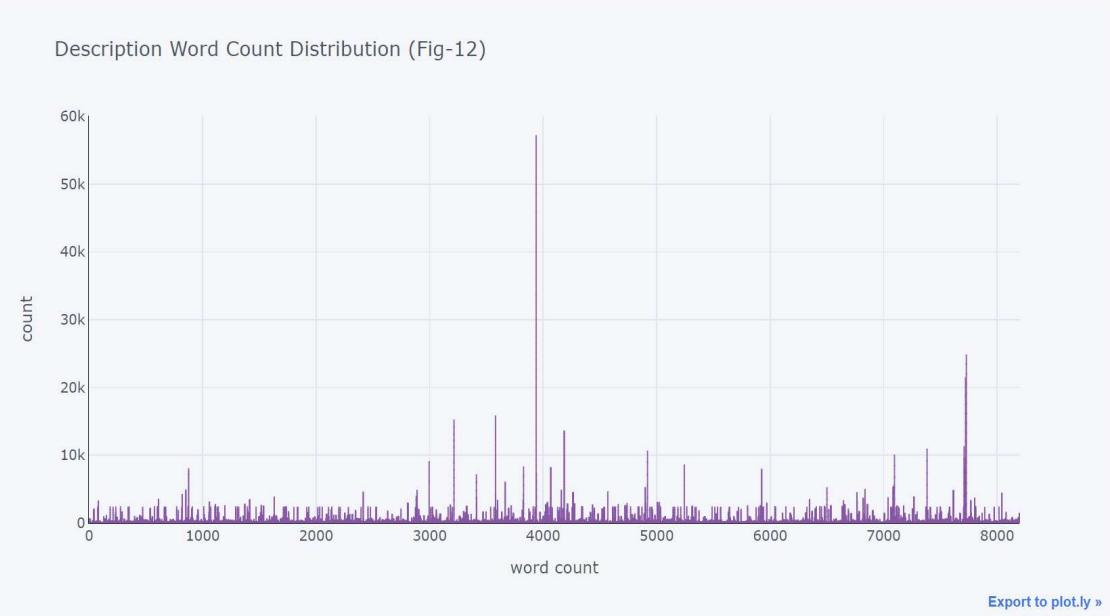
In [119]:
```
clean_data.insert(1, 'desc_len', clean_data['Final_Text'].astype(str).apply(len))
clean_data.insert(5, 'desc_word_count', clean_data['Final_Text'].apply(lambda x: len(str(x).split())))
clean_data.head()
```

Out[119]:

	Caller	desc_len	Assignment group	language	Final_Text	desc_word_count
3574	spxqmiry zpwgoqju	176	GRP_1	en	HostName _ : volume : / dev / ora_data encoura...	35
1590	jyoqwxhz clhxsoqy	185	GRP_1	en	HostName _ : volume : /dev / SID_ora server : ...	38
4783	pvlxjizg xzvlwqjc	487	GRP_1	en	hostname _ hostname _ listener active ever try...	80
541	kbnfxpsy gehxzayq	305	GRP_1	en	consider ticket_no correspond fix chg , key us...	47
4454	mnlazfsr mtqrkhnx	124	GRP_1	en	's shop_floor_app intelligence : inaccessible ...	17

In [120]:
```
# Description text Length
clean_data['desc_len'].iplot(
    kind='bar',
    xTitle='text length',
    yTitle='count',
    colorscale='-ylgn',
    title='Description Text Length Distribution (Fig-11)')

# Description word count
clean_data['desc_word_count'].iplot(
    kind='bar',
    xTitle='word count',
    linecolor='black',
    yTitle='count',
    colorscale='-bupu',
    title='Description Word Count Distribution (Fig-12)')
```





N-Grams

N-gram is a contiguous sequence of N items from a given sample of text or speech, in the fields of computational linguistics and probability. The items can be phonemes, syllables, letters, words or base pairs according to the application. N-grams are used to describe the number of words used as observation points, e.g., unigram means singly-worded, bigram means 2-worded phrase, and trigram means 3-worded phrase.

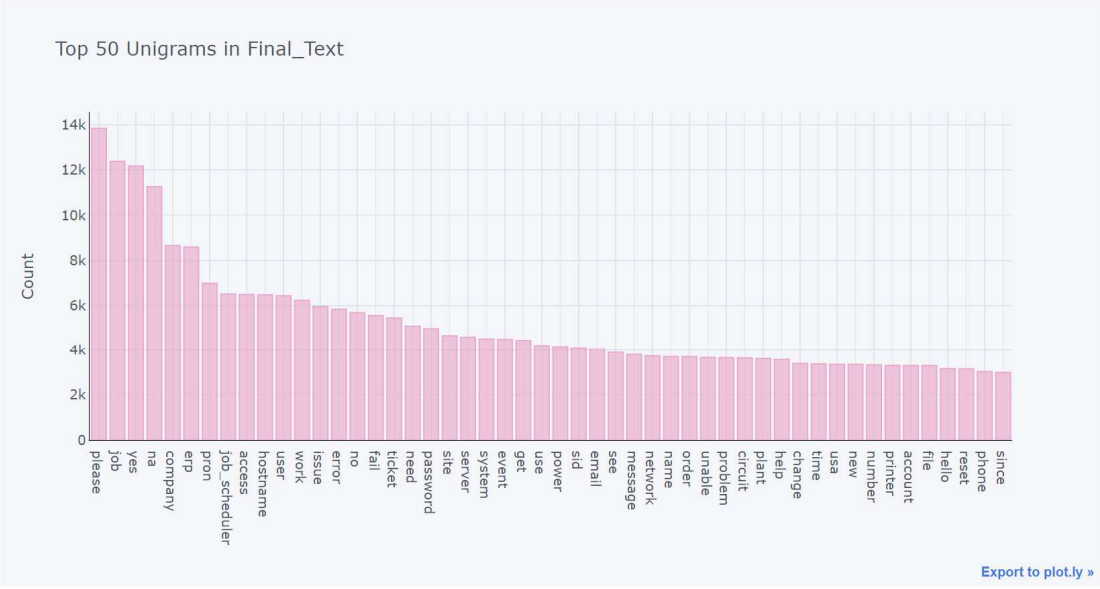
We'll be using scikit-learn's CountVectorizer function to derive n-grams and compare them before and after removing stop words. Stop words are a set of commonly used words in any language. We'll be using english corpus stopwords and extend it to include some business specific common words considered to be stop words in our case.

In [121]:
```
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from sklearn.feature_extraction.text import CountVectorizer

# Extend the English Stop Wordss
STOP_WORDS = STOPWORDS.union({'yes','na','hi',
                              'receive','hello',
                              'regards','thanks',
                              'from','greeting',
                              'forward','reply',
                              'will','please',
                              'see','help','able'})

# Generic function to derive top N n-grams from the corpus
def get_top_n_ngrams(corpus, top_n=None, ngram_range=(1,1), stopwords=None):
    vec = CountVectorizer(ngram_range=ngram_range,
                          stop_words=stopwords).fit(corpus)
    bag_of_words = vec.transform(corpus)
    sum_words = bag_of_words.sum(axis=0)
    words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
    words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)
    return words_freq[:top_n]
```

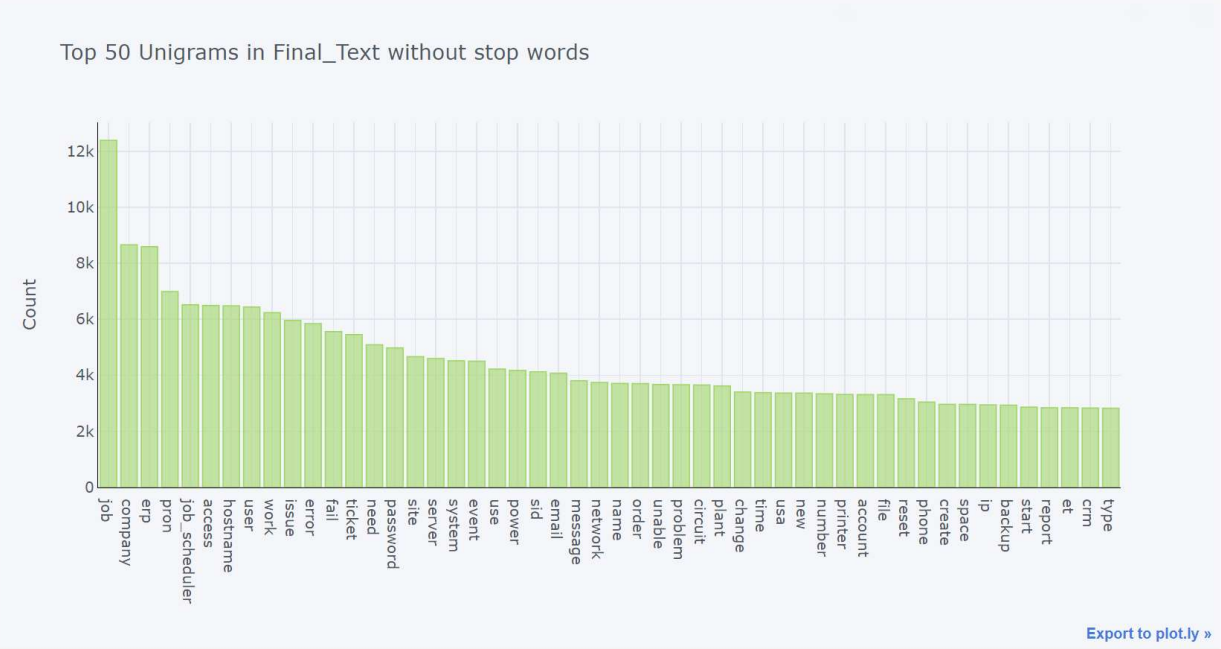
greatlearning
Learning for Life

Top Unigrams

In [122]:
```
# Top 50 Unigrams before removing stop words
top_n = 50
ngram_range = (1,1)
uni_grams = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range)

df = pd.DataFrame(uni_grams, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='piyg',
    title=f'Top {top_n} Unigrams in Final_Text')

# Top 50 Unigrams after removing stop words
uni_grams_sw = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range, stopwords=STOP_WORDS)

df = pd.DataFrame(uni_grams_sw, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='-piyg',
    title=f'Top {top_n} Unigrams in Final_Text without stop words')
```



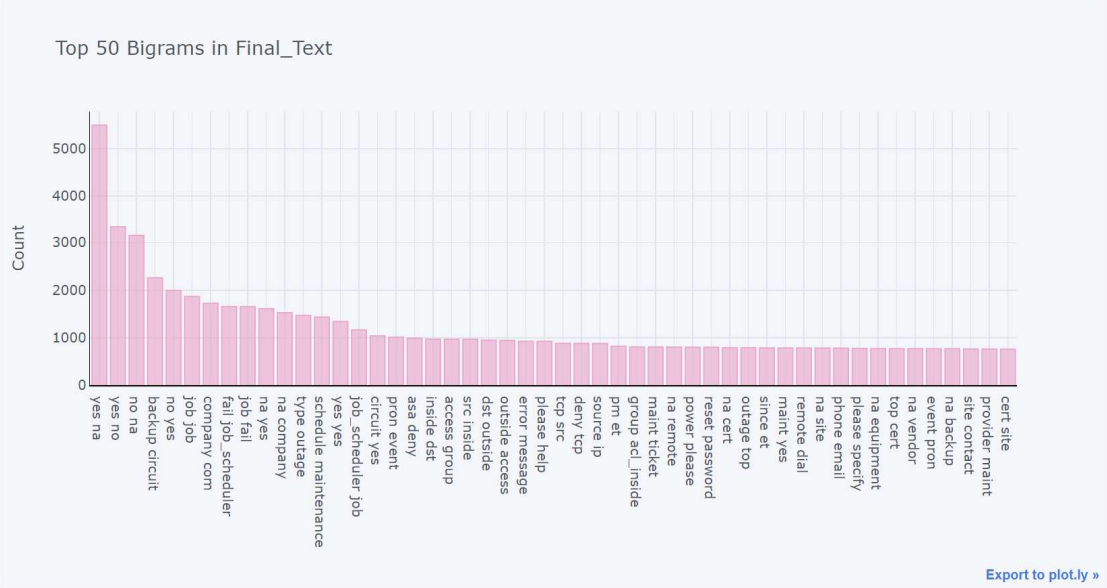

```
/usr/local/lib/python3.6/dist-packages/sklearn/feature_extraction/text.py:385: UserWarning:
```

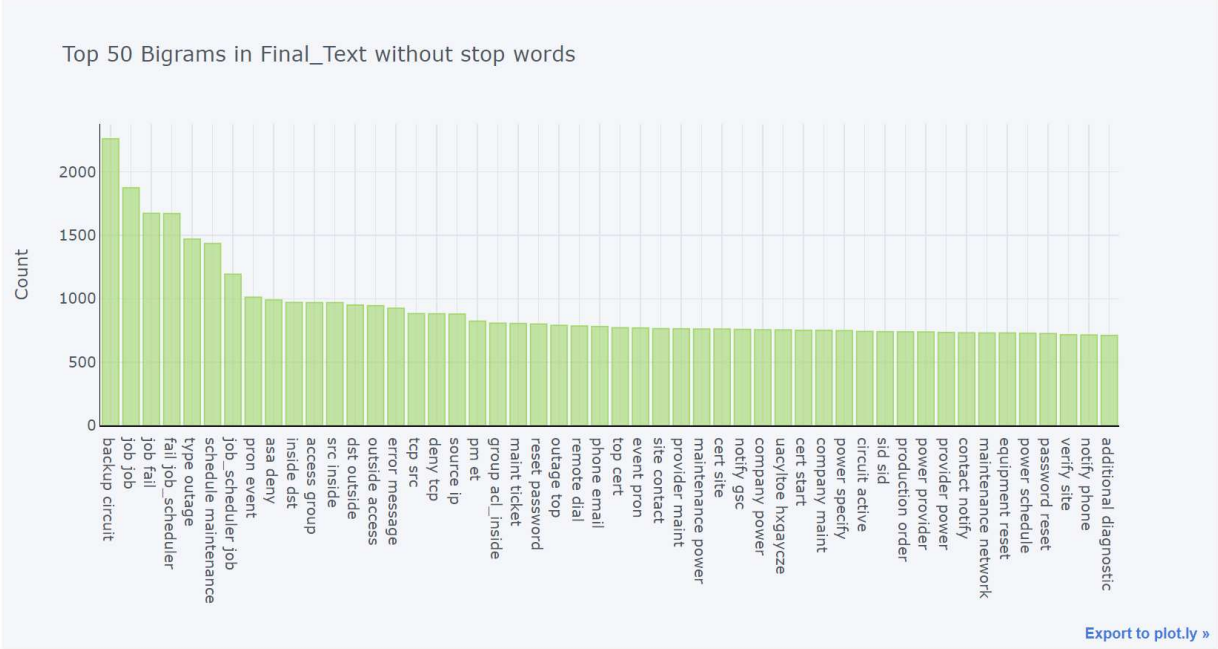

Top 50 Unigrams in Final_Text without stop words
Count
12k
10k
8k
6k
4k
2k
0
job
company
erp
pron
job_scheduler
access
hostname
user
work
issue
error
fail
ticket
need
password
site
server
system
event
use
power
sid
email
message
network
name
order
unable
problem
circuit
plant
change
time
usa
new
number
printer
account
file
reset
phone
create
space
ip
backup
start
report
et
crm
type
Export to plot.ly »

greatlearning
Learning for Life

Top Bigrams ¶

In [123]:
```
# Top 50 Bigrams before removing stop words
top_n = 50
ngram_range = (2,2)
bi_grams = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range)

df = pd.DataFrame(bi_grams, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='piyg',
    title=f'Top {top_n} Bigrams in Final_Text')

# Top 50 Bigrams after removing stop words
bi_grams_sw = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range, stopwords=STOP_WORDS)

df = pd.DataFrame(bi_grams_sw, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='-piyg',
    title=f'Top {top_n} Bigrams in Final_Text without stop words')
```

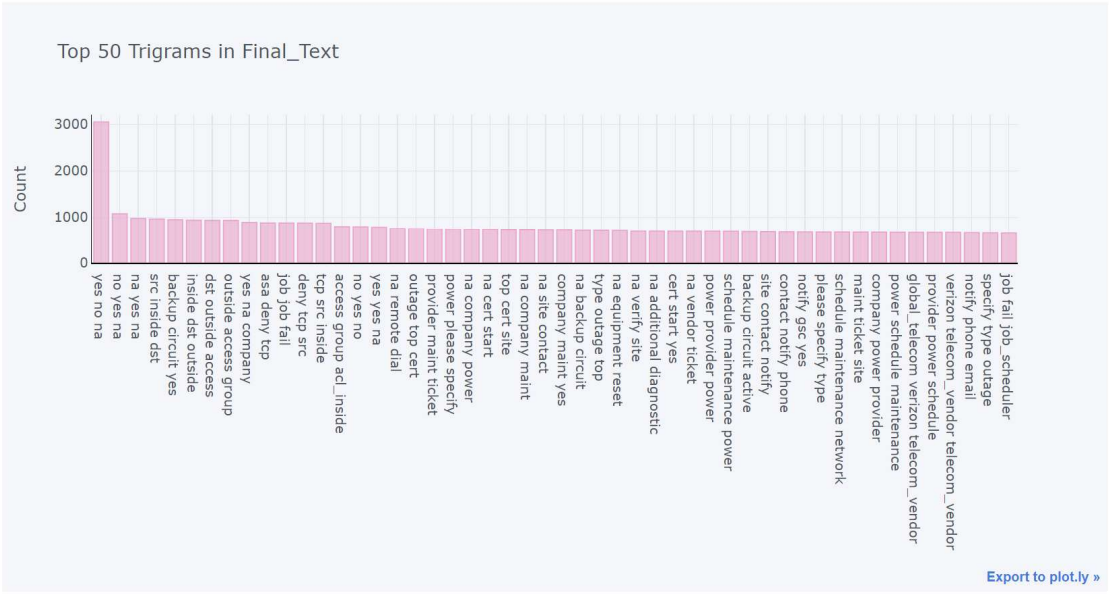


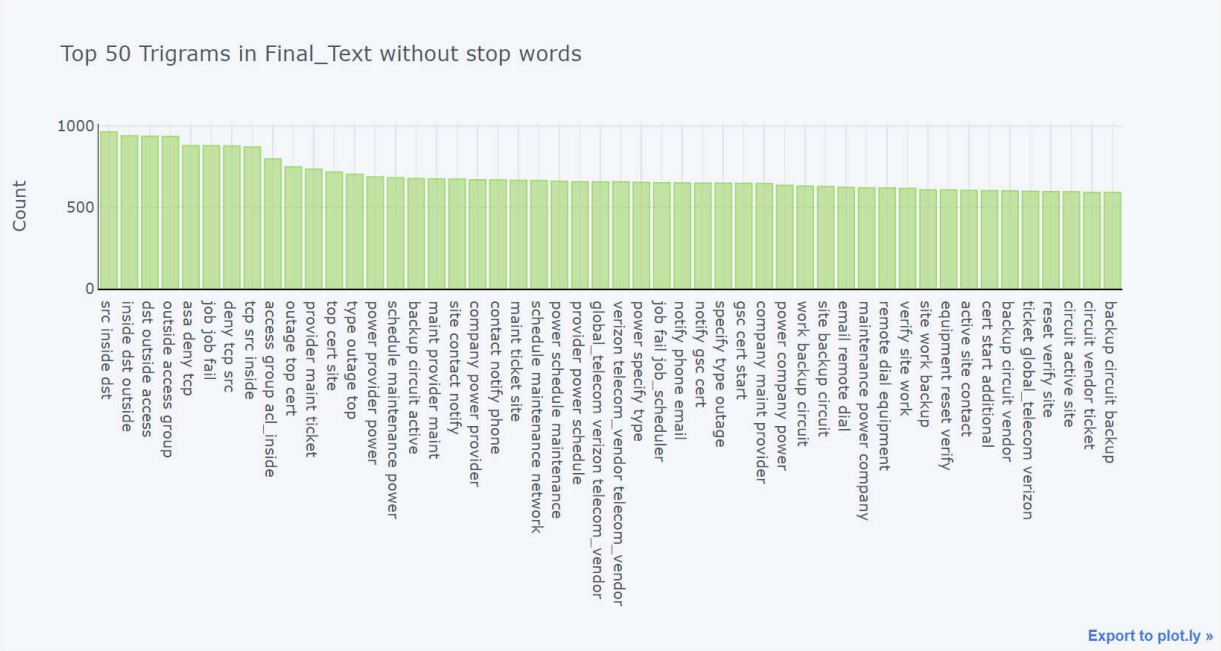

Top 50 Bigrams in Final_Text without stop words
Count
2000
1500
1000
500
0
backup circuit
job job
job fail
fail job_scheduler
type outage
schedule maintenance
job_scheduler job
pron event
asa deny
inside dst
access group
src inside
dst outside
outside access
error message
tcp src
deny tcp
source ip
pm et
group acl_inside
maint ticket
reset password
outage top
remote dial
phone email
top cert
event pron
site contact
provider maint
maintenance power
cert site
notify gsc
company power
uacyltoe hxgaycze
cert start
company maint
power specify
circuit active
sid sid
production order
power provider
provider power
contact notify
maintenance network
equipment reset
power schedule
password reset
verify site
notify phone
additional diagnostic
Export to plot.ly »

greatlearning
Learning for Life

Top Trigrams

In [124]:
```
# Top 50 Trigrams before removing stop words
top_n = 50
ngram_range = (3,3)
tri_grams = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range)

df = pd.DataFrame(tri_grams, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='piyg',
    title=f'Top {top_n} Trigrams in Final_Text')

# Top 50 Trigrams after removing stop words
tri_grams_sw = get_top_n_ngrams(clean_data.Final_Text, top_n, ngram_range, stopwords=STOP_WORDS)

df = pd.DataFrame(tri_grams_sw, columns = ['Final_Text' , 'count'])
df.groupby('Final_Text').sum()['count'].sort_values(ascending=False).iplot(
    kind='bar',
    yTitle='Count',
    linecolor='black',
    colorscale='-piyg',
    title=f'Top {top_n} Trigrams in Final_Text without stop words')
```





Word Cloud

Let us attempt to visualize this as a word cloud for top three groups that has got maximum records. A word cloud enables us to visualize the data as cluster of words and each words displayed in different font size based on the number of occurences of that word . Basically; the bolder and bigger the word show up in the visualization, it implies its more often it's mentioned within a given text compared to other words in the cloud and therefore would be more important for us.

Let's write a generic method to generate Word Clouds for both Short and Long Description columns.

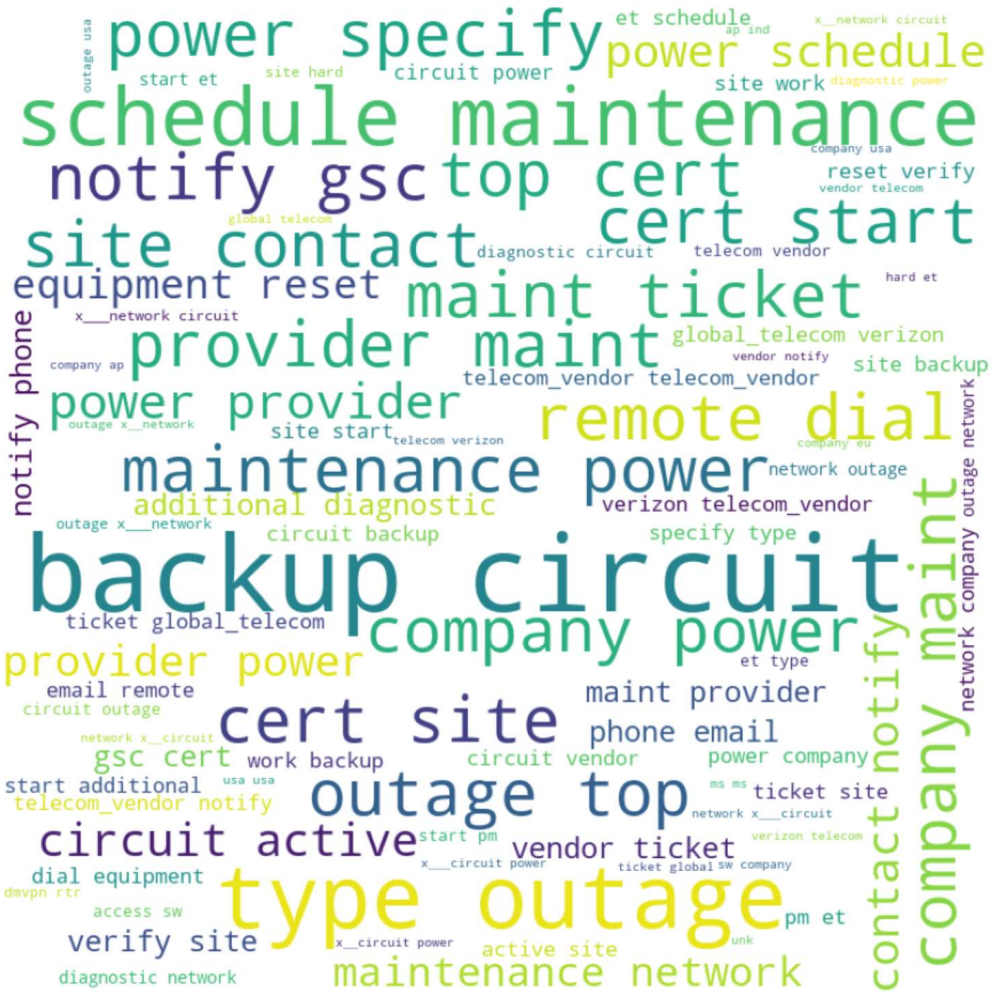
In [126]:
```
def generate_word_cloud(corpus):
        # Instantiate the wordcloud object
    wordcloud = WordCloud(width = 800, height = 800,
                    background_color ='white',
                    stopwords=STOP_WORDS,
                    # mask=mask,
                    min_font_size = 10).generate(corpus)

    # plot the WordCloud image
    plt.figure(figsize = (12, 12), facecolor = None)
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.tight_layout(pad = 0)

    plt.show()
```

greatlearning
Learning for Life

In [127]:
```
# Word Cloud for all tickets assigned to GRP_0
generate_word_cloud(' '.join(clean_data[clean_data['Assignment group'] == 'GRP_0'].Final_Text.str.strip()))
```

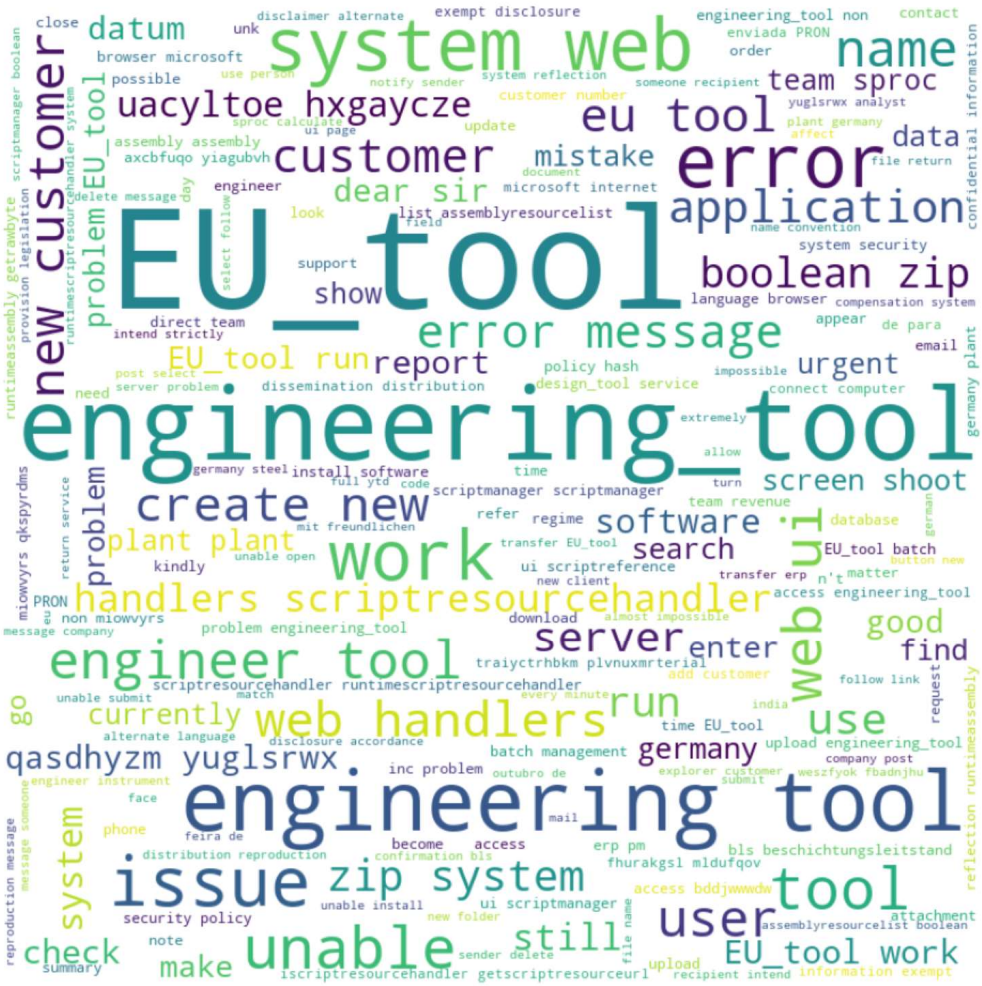
greatlearning
Learning for Life

In [128]:
```
# Word Cloud for all tickets assigned to GRP_8
generate_word_cloud(' '.join(clean_data[clean_data['Assignment group'] == 'GRP_8'].Final_Text.str.strip()))
```



greatlearning
Learning for Life

In [129]:
```
# Word Cloud for all tickets assigned to GRP_25
generate_word_cloud(' '.join(clean_data[clean_data['Assignment group'] == 'GRP_25'].Final_Text.str.strip()))
```



```
In [130]: # Generate wordcloud for Final_Text field
          generate_word_cloud(' '.join(clean_data.Final_Text.str.strip()))
```

- Then we proceeded ahead to preparing the data for model evaluation which is further detailed in the next section. “5. Model Evaluation”

5. Model evaluation

Describe the final model in detail. What was the objective, what parameters were prominent, and how did you evaluate the success of your models?

6. Comparison to benchmark

How does your final solution compare to the benchmark you laid out at the outset? Did you improve on the benchmark? Why or why not?

7. Visualizations

In addition to quantifying your model and the solution, please include all relevant visualizations that support the ideas/insights that you gleaned from the data.

8. Implications

How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?

9. Limitations

Few opportunities that we see and would like to pursue are as below.

- We have grouped the monitoring tool originated incidents that were scattered across different group with no meaningful /relatable information into single group. If we get an opportunity to further finetune this; we would like to explore that.
- We would like to revisit DNN scores and get more confidence on the model results; given the observation that we have.
- We would like to further explore whether we can improve the accuracy and Recall of Assignment Group_0.
- We further look forward to attempt BERT and Attention model.
- We would also like to further do Code Optimization given a chance.

10. Closing Reflections

What have you learned from the process? What you do differently next time?

Below are some of our learning from the project.

- The world of NLP is very vast; and we shall explore and understand alternate packages and methods than sticking on to one.
- The more time we spend in understanding the data helps us to get better end results.
- You will likely never have a perfectly balanced real-life data.
- We learnt not to make any assumptions while doing data pruning. That's definitely something that we will keep in mind next time.
- Working together in the project and contributing to overall solution was challenging as well as very exciting; given we had to keep in mind dependencies; and got to share varying perspectives and explore parallelly.
- While our benchmark was quite low; we constantly reassured each other not to get bogged down by numbers and keep the focus in learning and finding opportunities; at the same time, we challenged our own assumptions and results.

11. Acknowledgement

We would like to thank our mentor, **Mr. Sumit Kumar** who has guided us through this project and encouraged and challenged us to make iterative improvements. Also, this journey ;taking baby steps to the world of AI and ML would not have been possible without the support of the **Great Learning Faculty, Mentor Mr. Sathiya Kailas** and **Mentor Mr. Sumit Kumar**; who laid foundation taking us forward and our Program In charge **Miss**. **Ramya Nair** who has been very supportive and encouraging. We would like to extend our heartfelt gratitude to each one of them and **our families** who have motivated and inspired us; as we are making our final project submission.