AUTOMATIC TICKET ASSIGNMENT

Asjad, Kamal, Lakshmi, Pankaj

Contents

The Real Problem

One of the key activities of any IT function is to "Keep the lights on" to ensure there is no impact to the Business operations. IT leverages Incident Management process to achieve the above Objective. An incident is something that is unplanned interruption to an IT service or reduction in the quality of an IT service that affects the Users and the Business. The main goal of Incident Management process is to provide a quick fix / workarounds or solutions that resolves the interruption and restores the service to its full capacity to ensure no business impact. In most of the organizations, incidents are created by

various Business and IT Users, End Users/ Vendors if they have access to ticketing systems, and from the integrated monitoring systems and tools. Assigning the incidents to the appropriate person or unit in the support team has critical importance to provide improved user satisfaction while ensuring better allocation of support resources. The assignment of incidents to appropriate IT groups is still a manual process in many of the IT organizations. Manual assignment of incidents is time consuming and requires human efforts. There may be mistakes due to human errors and resource consumption is carried out ineffectively because of the misaddressing. On the other hand, manual assignment increases the response and resolution times which result in user satisfaction deterioration / poor customer service.

Business Domain Value

In the support process, incoming incidents are analyzed and assessed by organization's support teams to fulfill the request. In many organizations, better allocation and effective usage of the valuable support resources will directly result in substantial cost savings. Currently the incidents are created by various stakeholders (Business Users, IT Users and Monitoring Tools) within IT Service Management Tool and are assigned to Service Desk teams (L1 / L2 teams). This team will review the incidents for right ticket categorization, priorities and then carry out initial diagnosis to see if they can resolve. Around ~54% of the incidents are resolved by L1 / L2 teams. Incase L1 / L2 is unable to resolve, they will then escalate / assign the tickets to Functional teams from Applications and Infrastructure (L3 teams). Some portions of incidents are directly assigned to L3 teams by either Monitoring tools or Callers / Requestors. L3 teams will carry out detailed diagnosis and resolve the incidents. Around ~56% of incidents are resolved by Functional / L3 teams. Incase if vendor support is needed, they will reach out for their support towards incident closure. L1 / L2 needs to spend time reviewing Standard Operating Procedures (SOPs) before assigning to Functional teams (Minimum ~25-30% of incidents needs to be reviewed for SOPs before ticket assignment). 15 min is being spent for SOP review for each incident. Minimum of ~1 FTE effort needed only for incident assignment to L3 teams. Proprietary content. During the process of incident assignments by L1 / L2 teams to functional groups, there were multiple instances of incidents getting assigned to wrong functional groups. Around ~25% of Incidents are wrongly assigned to functional teams. Additional effort needed for Functional teams to re-assign to right functional groups. During this process, some of the incidents are in queue and not addressed timely resulting in poor customer service. Guided by powerful AI techniques that can classify incidents to right functional groups can help organizations to reduce the resolving time of the issue and can focus on more productive tasks.

Summary of problem statement, data, and findings

Automatic Ticket Assignment (ATA) is a classification problem which comes under the Supervised Machine Learning category & plays a key role for successfully running any Incident Management System, especially in very large system that provides numerous services, and each service has multiple categories and sub-categories. Manually tagging of task to specific category and sub-category requires user training, manpower and also prone to human error that can impact over all service delivery. ATA uses machine learning technique to assign task to appropriate group automatically that can improve overall turnaround time of service delivery.

Other business use case of text classification

(1) categorize Code review comments so that patterns of review comments can be identified and automated

(2) Post incident resolution in incident management system like SNOW, a user has to tag resolution comments to certain category. For e.g. in software incidents these categories may be (code issue, environment issue, Auto resolved, user training issue etc). Most often user misses to tag the comments to appropriate category. We can automate this process by creating model that can predict appropriate category for any resolution comment.

(3) After sales support in product based companies, assignment of correct service personnel so that cost can be optimized and customer satisfaction can be enhanced

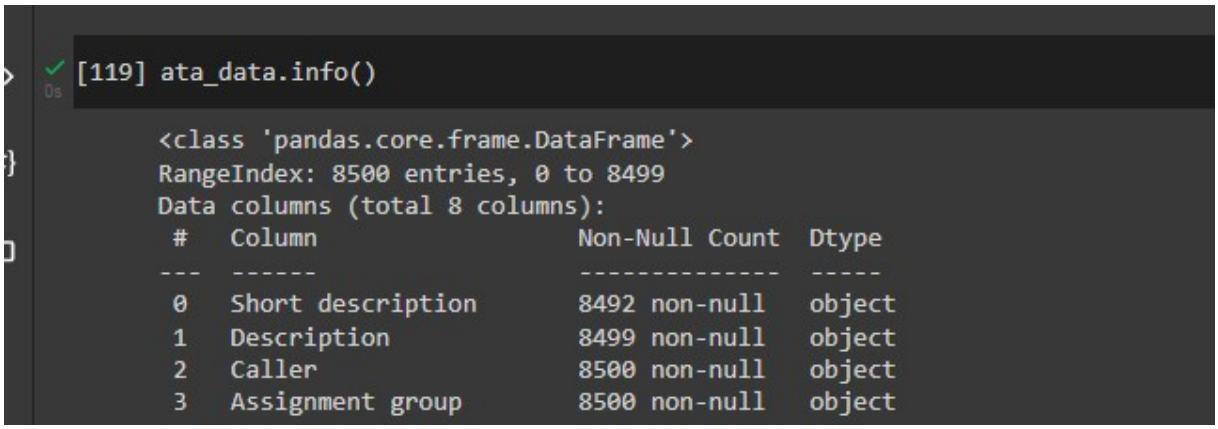
Data Shape

There are 8,500 rows and 4 columns in the base data set

```
[46] ata_data.shape

     (8500, 4)
```

Some more info on the dataset

The four columns are:-



```
[119] ata_data.info()

      <class 'pandas.core.frame.DataFrame'>
      RangeIndex: 8500 entries, 0 to 8499
      Data columns (total 8 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
```

Quick peek into data

```
[44] ata_data.head()
```

	Short description	Description	Caller	Assignment group
0	login issue	-verified user details.(employee# & manager name)\r\n-checked the user name in ad and reset the password.\r\n-advised the user to login and check.\r\n-caller confirmed that he was able to login.\r\n-issue resolved.	spxjnwir pjlcoqds	GRP_0
1	outlook	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail.com\r\n\r\nhello team,\r\n\r\nmy meetings/skype meetings etc are not appearing in my outlook calendar, can somebody please advise how to correct this?\r\n\r\nkind	hmjdrvpb komuaywn	GRP_0
2	cant log in to vpn	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail.com\r\n\r\nhi\r\n\r\ni cannot log on to vpn\r\n\r\nbest	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

```
[45] ata_data.tail()
```

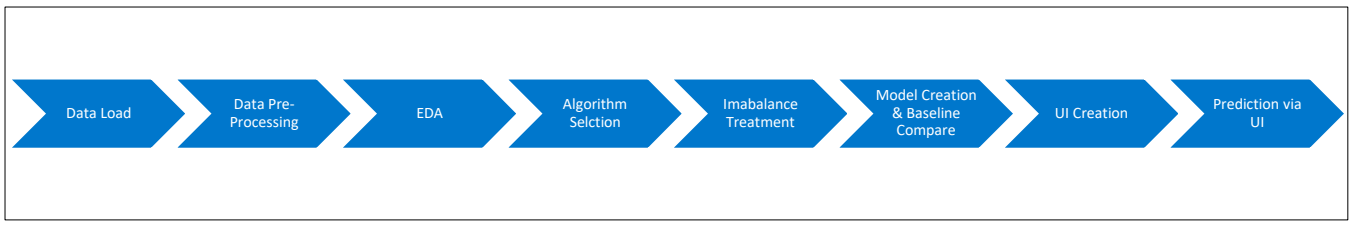
	Short description	Description	Caller	Assignment group
8495	emails not coming in from zz mail	\r\n\r\nreceived from: avglmrts.vhqmtiua@gmail.com\r\n\r\ngood afternoon,\r\ni am not receiving the emails that i sent from zz mail.\r\nplease advise\r\n\r\n	avglmrts vhqmtiua	GRP_29
8496	telephony_software issue	telephony_software issue	rbozivdq gmlhrtvp	GRP_0
8497	vip2: windows password reset for tifpdchb pedxruyf	vip2: windows password reset for tifpdchb pedxruyf	oybwdsgx oxyhwrtz	GRP_0
8498	machine nÃ£o estÃ¡ funcionando	i am unable to access the machine utilities to finish the drawers adjustment settings.\r\nis no network..	ufawcgob aowhxjky	GRP_62
8499	an mehreren pc`s lassen sich verschiedene prgramdntyme nicht Ã¶ffnen.	an mehreren pc`s lassen sich verschiedene prgramdntyme nicht Ã¶ffnen. bereich cnc.	kqvbrspl jyzokltx	GRP_49

Initial impression of data

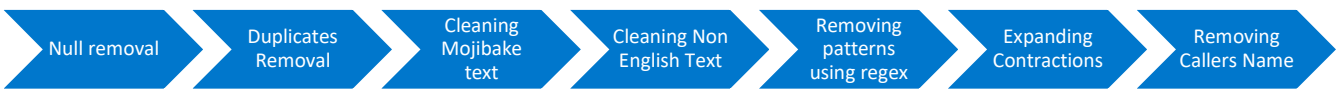
- Presence of
  - null values
  - duplicate values
  - mojibake text
  - text in different languages
  - highly imbalanced dataset

Overview of the final process

Overall process flow



Data pre-processing steps



Step by Step Walkthrough the solution

Challenges faced in cleaning data

Check for nulls

8 values in Short Description and 1 in Description are null values

```
[47] ata_data.isnull().sum()

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

There are some null values in Short Description & Description column and putting them in a different column for analysis purpose

```
ata_data[ata_data['Description'].isnull()]
```

	Short description	Description	Caller	Assignment group
4395	i am locked out of skype	NaN	viyglzfo ajtfzpkb	GRP_0

```
[48] ata_data[ata_data['Short description'].isnull()]
```

	Short description	Description	Caller	Assignment group
2604	NaN	\r\n\r\nreceived from: ohdrnswl.rezuibdt@gmail.com\r\n\r\nhi,\r\n\r\n\r\n\r\nlink is not working. kindly resolve yhe issue on urgent basis.\r\n\r\n\r\n\r\nbest	ohdrnswl rezuibdt	GRP_34
3383	NaN	\r\n-connected to the user system using teamviewer.\r\n-help the user login to the portal.\r\n-issue resolved.	qftpazns fxpnytmk	GRP_0
3906	NaN	-user unable tologin to vpn.\r\n-connected to the user system using teamviewer.\r\n-help the user login to the company vpn using the vpn company vpn link.\r\n-issue resolved.	awpcmsey cldiuqwe	GRP_0
3910	NaN	-user unable tologin to vpn.\r\n-connected to the user system using teamviewer.\r\n-help the user login to the company vpn using the vpn company vpn link.\r\n-issue resolved.	rhwsmefo tvphyura	GRP_0
3915	NaN	-user unable tologin to vpn.\r\n-connected to the user system using teamviewer.\r\n-help the user login to the company vpn using the vpn company vpn link.\r\n-issue resolved.	hxripljo efzounig	GRP_0
3921	NaN	-user unable tologin to vpn.\r\n-connected to the user system using teamviewer.\r\n-help the user login to the company vpn using the vpn company vpn link.\r\n-issue resolved.	cziadygo veiosxby	GRP_0
3924	NaN	name:wvqgbdhm fwchqjor\nlanguage:\nbrowser:microsoft internet explorer\nemail:wvqgbdhm.fwchqjor@gmail.com\ncustomer number:\ntelephone:-not available\nsummary:can't get into vpn - need to be on at 4:30 est and it wont' happen please help aerpf	wvqgbdhm fwchqjor	GRP_0
4341	NaN	\r\n\r\nreceived from: eqmuniov.ehxkcbgj@gmail.com\r\n\r\ngood morning,\r\n\r\nwhen trying to log on to erp i get this error below.\r\nplease help urgently as i can not process customer order.\r\n\r\n[cid:image001.png@01d20f2e.751db880]\r\n\r\n\r\n	eqmuniov ehxkcbgj	GRP_0

Handling Nulls

Wherever short description is null, replacing it with description and vice versa. The data is put in another column short_desc_analysis and desc_analysis which will be used for further analysis. The base columns are kept intact

```
[51] ata_data['short_desc_analysis'] = np.where(ata_data['Short description'].isnull(), ata_data['Description'], ata_data['Short description'])
     ata_data['desc_analysis'] = np.where(ata_data['Description'].isnull(), ata_data['Short description'], ata_data['Description'])

[52] ata_data.head()
```

	Short description	Description	Caller	Assignment group	short_desc_analysis	desc_analysis
0	login issue	-verified user details.(employee# & manager name)\r\n-checked the user name in ad and reset the password.\r\n-advised the user to login and check.\r\n-caller confirmed that he was able to login.\r\n-issue resolved.	spxjnwir pjlcoqds	GRP_0	login issue	-verified user details.(employee# & manager name)\r\n-checked the user name in ad and reset the password.\r\n-advised the user to login and check.\r\n-caller confirmed that he was able to login.\r\n-issue resolved.
1	outlook	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail.com\r\n\r\nhello team,\r\n\r\nmy meetings/skype meetings etc are not appearing in my outlook calendar, can somebody please advise how to correct this?\r\n\r\nkind	hmjdrvpb komuaywn	GRP_0	outlook	\r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail.com\r\n\r\nhello team,\r\n\r\nmy meetings/skype meetings etc are not appearing in my outlook calendar, can somebody please advise how to correct this?\r\n\r\nkind
2	cant log in to vpn	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail.com\r\n\r\nhi\r\n\r\ni cannot log on to vpn\r\n\r\nbest	eylqgodm ybqkwiam	GRP_0	cant log in to vpn	\r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail.com\r\n\r\nhi\r\n\r\ni cannot log on to vpn\r\n\r\nbest
3	unable to access hr_tool page	unable to access hr_tool page	xbkucsvz gcpydteq	GRP_0	unable to access hr_tool page	unable to access hr_tool page
4	skype error	skype error	owlgqjme qhcozdfx	GRP_0	skype error	skype error

```
[53] ata_data.isnull().sum()

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

Identifying Duplicates

```
# Finding duplicate records
ata_data[ata_data.duplicated()]['Short description'].value_counts()
```

```
blank call                                                                                  10
call for ecwtrjnq jpecxuty                                                                   6
reset passwords for bxeagsmt zrwdgsco using password_management_tool password reset.         5
job Job_3028 failed in job_scheduler at: 08/24/2016 00:02:00                                 4
blank call //gso                                                                             3
call came and got disconnected                                                               3
blank call // loud noise                                                                     2
job Job_1314 failed in job_scheduler at: 08/25/2016 08:15:00                                 2
german call                                                                                  2
svc-now ticket found... doing nothing                                                        2
```

Removing duplicates

After removing duplicates remaining records: 8417

```
ata_data=ata_data.drop_duplicates()
```

```
ata_data.duplicated().value_counts()
```

```
False    8417
dtype: int64
```

Mojibake

The base data has presence of scrambled text called Mojibake. Example given below. It occurs when we try to read text in some other encodings

```
('ç"µè„'ç™»å¼•å¯†ç ⊡å¿˜è®°ï¼Œé‡⊡ç½®å¯†ç ⊡ã€,')
```

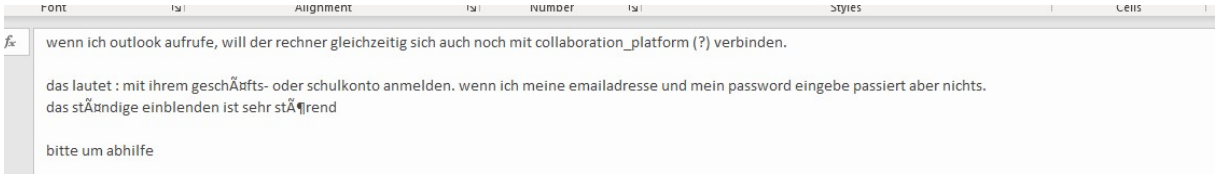
Reference: https://www.kaggle.com/rtatman/data-cleaning-challenge-character-encodings

Package FTFY is used to clean the Mojibake text. The below code snippet shows that Mojibake texts are indeed non-english text

```
[120] ftfy.fix_text('ç"µè„'ç™»å¼•å¯†ç ⊡å¿˜è®°ï¼Œé‡⊡ç½®å¯†ç ⊡ã€,')

      '电脑登录密码忘记,重置密码。 '
```

Presence of non-English language

Cleaning Mojibake text helps us understand that there are non-English texts. But apart from cleaned Mojibake as well, we can find non-English text in corpus



Google Translator is used to translate non-English text to English

```
[104] to_translate = ftfy.fix_text('aktuell kÃ¶nnen keine rÃ¼ckmeldungen in EU_tool eingegeben werden. fehler "laufzeitfehler".')
      translated = GoogleTranslator(source='auto', target='english').translate(to_translate)

      print(translated)

      No feedback can currently be entered in EU_tool. error "runtime error".
```

```
to_translate = ftfy.fix_text('ç”µè„‘ç™»å½•å¯†ç ä¸¢å¤±è¯·é‡ç½®å¯†ç ã€‚')
translated = GoogleTranslator(source='auto', target='english').translate(to_translate)

print(translated)

If the computer login password is forgotten, reset the password.
```

Checking for Patterns: "received from: *eylqgodm.ybqkwiam@gmail.com*"

The portion in italics can be any mail id. Further investigation showed there are 2251 such records out of 8500.

The pattern is removed from the text and the email id is added to another mail_received_from

```
ata_data.tail()
```

	Short description	Description	Caller	Assignment group	short_desc_analysis	desc_analysis	mail_received_from	mail_subject_mentioned
8495	emails not coming in from zz mail	\r\n\r\nreceived from: avglmrts.vhqmtiua@gmail.com\r\n\r\ngood afternoon,\r\ni am not receiving the emails that i sent from zz mail.\r\nplease advise\r\n\r\n	avglmrts vhqmtiua	GRP_29	emails not coming in from zz mail	i am not receiving the emails that i sent from zz mail. please advise	avglmrts.vhqmtiua@gmail.com	No Match
8496	telephony_software issue	telephony_software issue	rbozivdq gmlhrtvp	GRP_0	telephony_software issue	telephony_software issue	No Match	No Match
8497	vip2: windows password reset for tifpdchb pedxruyf	vip2: windows password reset for tifpdchb pedxruyf	oybwdsgx oxyhwrfz	GRP_0	vip: windows password reset for tifpdchb pedxruyf	vip: windows password reset for tifpdchb pedxruyf	No Match	No Match
8498	machine nÃ£o estÃ¡ funcionando	i am unable to access the machine utilities to finish the drawers adjustment settings.\r\nis no network..	ufawcgob aowhxjky	GRP_62	mace não está funcionando	i am unable to access the mace utilities to finish the drawers adjustment settings. is no network..	No Match	No Match

Analyzing the mails received from indicates that there are 961 records where initially issue was triggered due to system generated mail from monitoring_tool@company.com

```
[54] ata_data[ata_data['mail_received_from']!='No Match']['mail_received_from'].value_counts()

monitoring_tool@company.com    961
rxoynvgi.ntgdsehl@gmail.com     14
gjtyswkb.dpvaymxr@gmail.com     11
vkzwafuh.tcjnuswg@gmail.com     10
zuxcfonv.nyhpkrbe@gmail.com      9
                               ...
anivdcor.rbmfhiox@gmail.com      1
ebqdmgpk.daoyrtmj@gmail.com      1
qgrbdnoc.dgupnhxv@gmail.com      1
esaqztby.mhnbqiyc@gmail.com      1
znxcupyi.bhrwyxgu@gmail.com      1
Name: mail_received_from, Length: 711, dtype: int64
```

Checking for Patterns: "email ids"

A description could contain multiple email ids. Removing that pattern and replacing with blank

```
[31] def remove_email(text):
       matched_emails = re.findall('\S+@\S+', text)

       for email in matched_emails:
         text = text.replace(email,"")
       return text
```

Checking for Patterns: "Mail Format"

Some of the Description have pattern like that of mail

(1) From

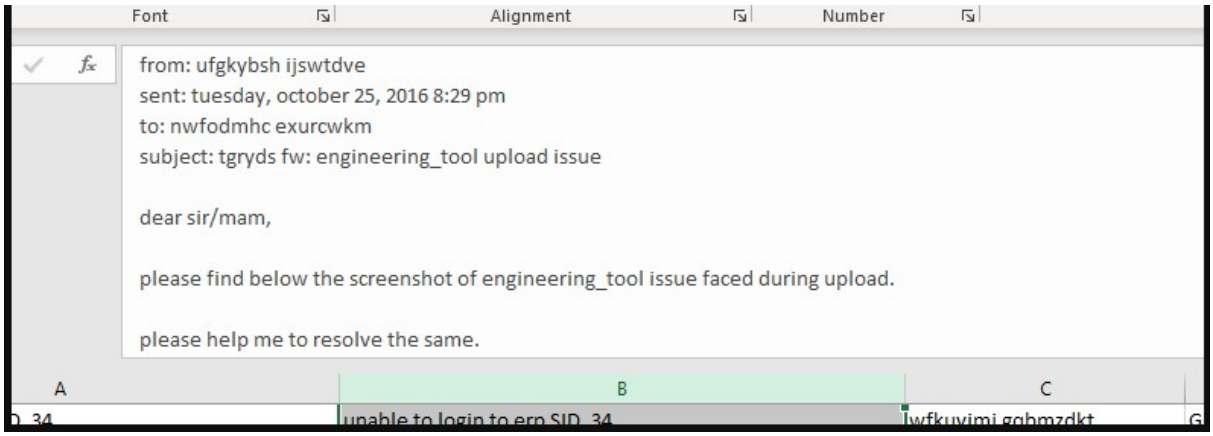
(2) Sent

(3) To

(4) Subject

(5) Cc

(6) importance



Removing such patterns using regex and replacing with blanks. Also the subject is copied into another column mail_subject_mentioned

```
match_from = re.findall(r'from:.*[\r\n]', text)
match_to = re.findall(r'to:.*[\r\n]', text)
match_sent = re.findall(r'sent:.*[\r\n]', text)
match_date = re.findall(r'date:.*[\r\n]', text)
match_cc = re.findall(r'cc:.*[\r\n]', text)
match_subject = re.findall(r'subject:.*[\r\n]', text)
match_importance = re.findall(r'importance:.*[\r\n]', text)
```

Clean text

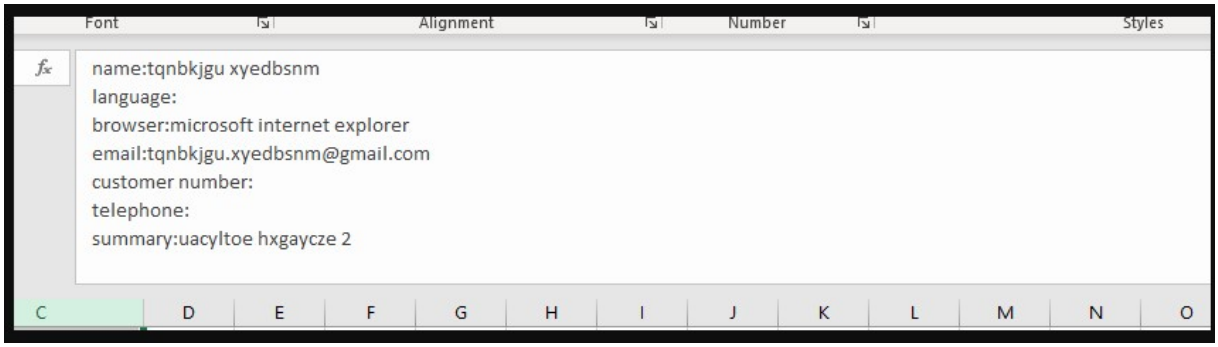
Checking for Patterns: <mailto:>

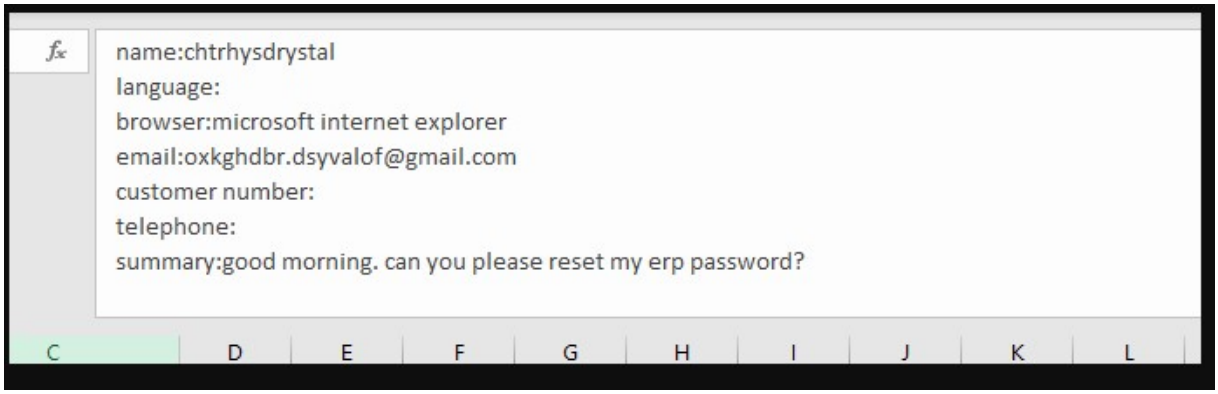
Removing occurrences of mailto: from the text

```
match_mailto = re.findall(r'<mailto:.*>', text)
```

Checking for Patterns: template with name, language, browser, etc.

The description column has a template for e.g.





Removing such pattern using regex

```
match_first_name = re.findall(r'first name.*[\r\n]', text)
match_last_name = re.findall(r'last name.*[\r\n]', text)
match_user_name_space = re.findall(r'user name:.*[\r\n]', text)
match_user_name = re.findall(r'username:.*[\r\n]', text)
match_name = re.findall(r'name:.*[\r\n]', text)
match_language = re.findall(r'language:.*[\r\n]', text)
match_browser = re.findall(r'browser:.*[\r\n]', text)
match_mail_id = re.findall(r'mail id:.*[\r\n]', text)
match_email_address = re.findall(r'email address:.*[\r\n]', text)
match_email = re.findall(r'email:.*[\r\n]', text)
match_customernumber = re.findall(r'customer number:.*[\r\n]', text)
match_customerjobtitle = re.findall(r'customer job title:.*[\r\n]', text)
match_telephone = re.findall(r'telephone:.*[\r\n]', text)
match_contact = re.findall(r'contact #:.*[\r\n]', text)
match_vit_ref_num = re.findall(r'vitalyst reference number:.*[\r\n]', text)
match_supervisor = re.findall(r'supervisor:.*[\r\n]', text)
match_manager = re.findall(r'manager.*[\r\n]', text)
match_i_number = re.findall(r'i number:.*[\r\n]', text)
match_cost_center = re.findall(r'cost center.*[\r\n]', text)
match_ext_comp = re.findall(r'external company name.*[\r\n]', text)
match_emp_id = re.findall(r'emp id:.*[\r\n]', text)
```

Checking for Patterns: Checking for embedded images text

The data contains reference of embedded images as shown in the below image

```
[58] ata_data[ata_data['Description'].str.contains('cid:image', case=False, na=False)].head()
```

	Short description	Description	Cal
21	vpn issue	\r\n\r\nreceived from: ugephfta.hrbqkvij@gmail.com\r\n\r\nhello helpdesk\r\n\r\ni am not able to connect vpn from home office. couple f hours ago i was connected, now it is not working anymore. getting a message that my session expired but if i click on the link, nothing happens.\r\n\r\n[cid:image001.jpg@01d233aa.3f618be0]\r\n\r\n**********\r\n\r\nneed help with your dynamics crm?\r\nclick here<\r\n\r\nchat with a live agent regarding your dynamics crm questions now! click here<\r\n\r\nbest	ugep hrbc
62	issues with outlook	\r\n\r\nreceived from: lkfzibrx.ljnabpgx@gmail.com\r\n\r\n[cid:image001.png@01d23357.fcbe58b0]\r\n\r\nbest	lkfz ljnab
107	attendance_tool - system log on error	\n\nreceived from: isfadulo.etkyjabn@gmail.com\n\nhello\n\ngood morning,\n\ni am experiencing issues with attendance_tool log on\nevery time i try to log on through "single sign portal, the following screen gets displayed and it stays there.\nappreciate your support to fix this issue\n\n\n[cid:image011.jpg@01d231cb.2cceacf0]\n\n	isfa etky
128	password change thru password_management_tool password manager	\r\n\r\nreceived from: jvpkulxw.ovuweygj@gmail.com\r\n\r\nhello sir,\r\n\r\ni tried to change my password thru above. i got below error. pl help know what action to be taken further to ensure all passwords are same everywhere since belo wmsg says all passwords were not changed.\r\n\r\n[cid:image001.jpg@01d2316b.5ff15980]\r\n\r\n	jvpku ovuw
151	i used to have acces to this location on collaboration_platform. now i do not. i need access.	\n\nreceived from: bwfhtumx.japznrvb@gmail.com\n\n\n\n[cid:image001.jpg@01d230f7.8bb4e830]\n\nbwfhtumx japznrvb \nregional controller\nbwfhtumx.japznrvb@gmail.com<mailto:bwfhtumx.japznrvb@gmail.com>\n\n\n\n	bwfht japz

Regex is used to replace such patterns with blanks

```
match_emb_image = re.findall(r'\[cid:image.*\]', text)
match_begin_fwd_msg = re.findall(r'begin forwarded message:', text)
```

Checking for Patterns: Phrases

Certain pattern of text found in corpus

- begin forwarded message:
- sent from my iphone
- sent from my ipad
- “sir or madam,” or “sir/mam,” or “sir,”
- yes/no/na
- good day or good afternoon or good morning or good evening
- hello i.t. team or hello help-team or hello support team or hello help-team or hello it-team or hello ladies and gentlemen or hello it helper or hellow or hello it support or hello all or hello colleagues or hi there or hello it team or hello sir or hello it service or hello it or hello helpdesk or hello team or hello all or hello it desk or hello it helper or hello dac or hello or gentles or it

team or dear all or dear it or dear or hallo or all groups or it help or team ith best or best or with kind or kind or many or with warm or warm

- hi it or hi team or hi it experts or hi
- thanking you or thanking u or thank u or thank you or thanks

Such patterns are removed using regex

```
[33] def remove_greetings(text):
       match_greetings = re.search(r'(^|\s)(good day|good afternoon|good morning|good evening)(,|\s|!|.|:)', text)
       if bool(match_greetings):
         text = text.replace(match_greetings.group(0),"")
       return text

[34] def remove_best_wishes(text):
       match_bestwishes = re.search(r'(with best|best|with kind|kind|many|with warm|warm)$', text)
       if bool(match_bestwishes):
         text = text.replace(match_bestwishes.group(0),"")
       return text

     def remove_hello_wishes(text):
       match_hellowishes = re.search(r'(hello i.t. team|hello help-team|hello support team|hello help-team|hello it-team|hello ladies and gentlemen|hello  it helper|hellow|hello it support|hello
       if bool(match_hellowishes):
         text = text.replace(match_hellowishes.group(0),"")
       return text

[36] def remove_hi(text):
       match_hi = re.search(r'(hi it|hi team|hi it experts|hi)(,|\s|!|.|:)', text)
       if bool(match_hi):
         text = text.replace(match_hi.group(0),"")
       return text

[37] def remove_thanking(text):
       match_thanking = re.search(r'(thanking you|thanking u|thank u|thank you|thanks)(,|.|/s|$)', text)
       if bool(match_thanking):
         text = text.replace(match_thanking.group(0),"")
       return text
```

Checking for Patterns: Expanding acronyms such as “pls” and replacing phrase such as “please help to”

Short form such as pls is replaced with please and then “please help to” is replaced with blanks

```
[38] def expand_pls(text):
       match_pls = re.search(r'(pls)(\s|.)', text)
       if bool(match_pls):
         text = text.replace(match_pls.group(0),"")
       return text

[39] def remove_please(text):
       match_help = re.search(r'please help to', text)
       if bool(match_help):
         text = text.replace(match_help.group(0),"")
       return text
```

Expanding contractions

Contractions such as isn’t, can’t, doesn’t can be expanded to is not, cannot & does not resp. This is done using the package **contractions**

```
[41] def fix_contractions(text):
       fixed_contractions = contractions.fix(text)
       return fixed_contractions
```

Applying contractions.fix

```
[117] text = '''i had to reset my password again and i've lost my option for setting up a skype meeting again.  can you please help me?  i can't recall how you were able to bring it back the last time.'''

      contractions.fix(text)

      'i had to reset my password again and I have lost my option for setting up a skype meeting again.  can you please help me?  i cannot recall how you were able to bring it back the last time.'
```

Removing Caller's name

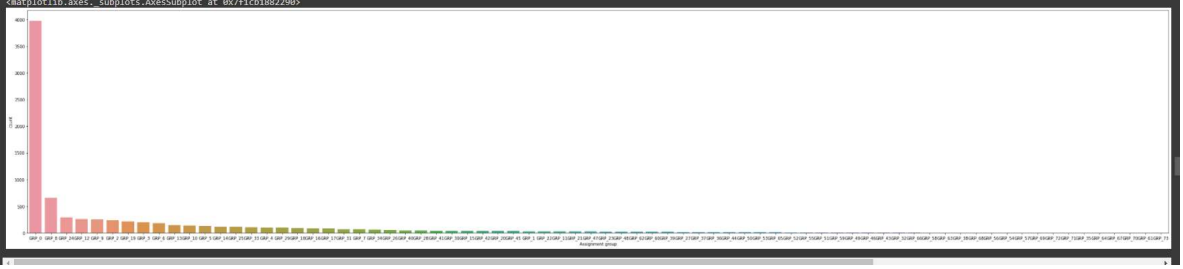
The caller's name is removed from short description and long description

```
Removing the Caller name from the description.

[ ]  caller_list = list(ata_data["Caller"].str.split(" ", expand = True)[0].unique()) + list(ata_data["Caller"].str.split(" ", expand = True)[1].unique())

     for name in caller_list:
       ata_data['short_desc_analysis'] = ata_data['short_desc_analysis'].replace(name,"")
       ata_data['desc_analysis'] = ata_data['desc_analysis'].replace(name,"")
```

Analysis of the data

Distribution of tickets as per Assignment Group

```
assignment_group_desc_order = ata_data['Assignment group'].value_counts().sort_values(ascending=False).index
plt.figure(figsize=(50,10))
sns.countplot(ata_data['Assignment group'], order=assignment_group_desc_order)

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other argument
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f1cb1882290>
```

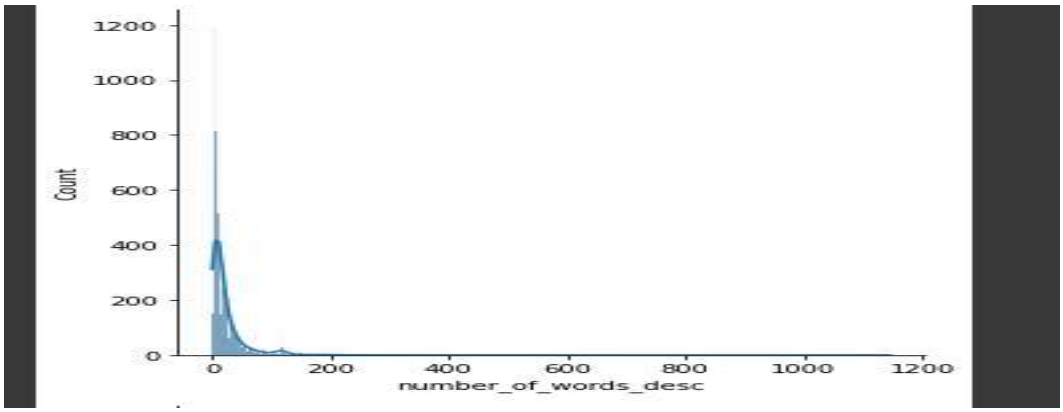


Group_0 has 3976 tickets (roughly 47% of tickets). Dataset is imbalanced

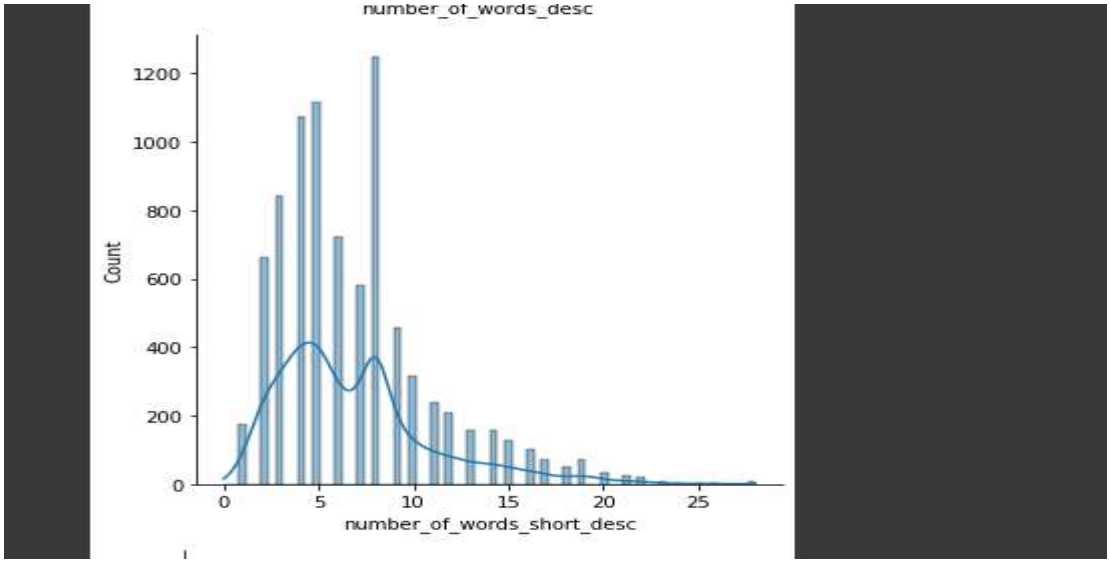
- no. of group less than 10 tickets- 25
- no. of groups in which records between 10 and 100 - 37
- no. of group greater than 100 tickets – 11

Words/Characters analysis

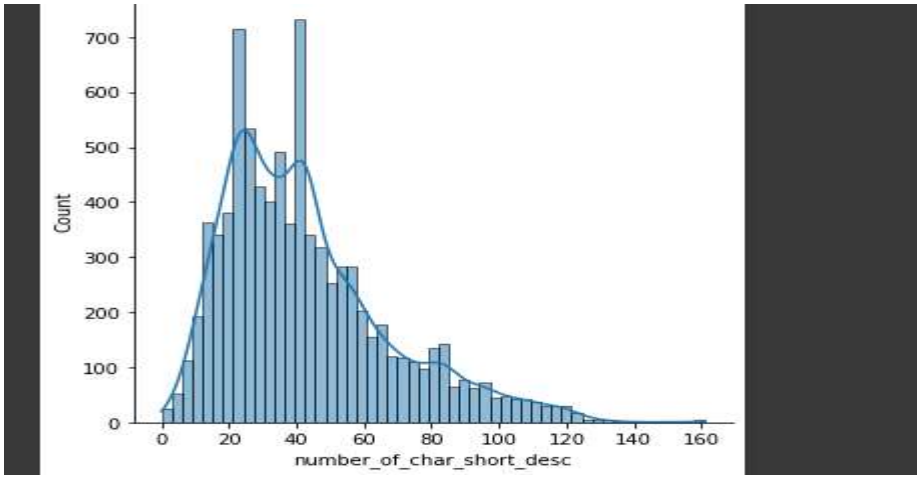
Majority of tickets have upto 100 words in Description. However, there are some tickets ranging from 200 – 1200 words



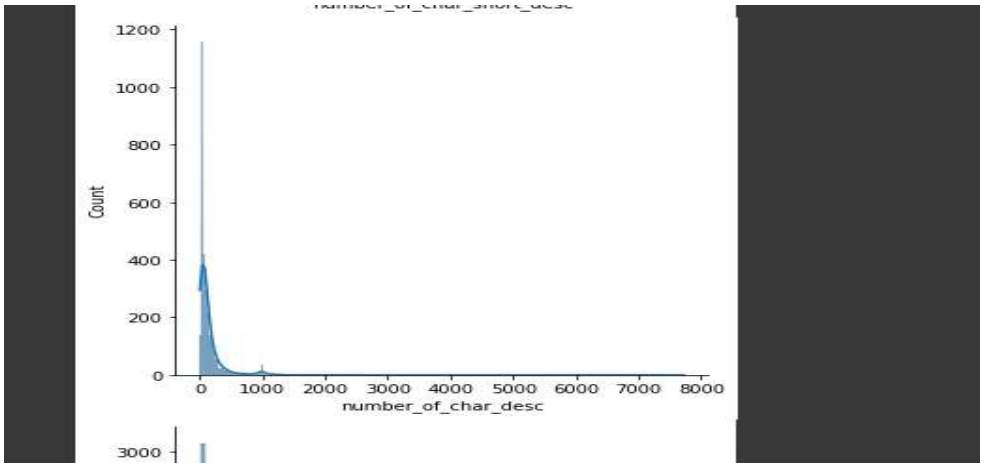
Short Description has a much compact word distribution



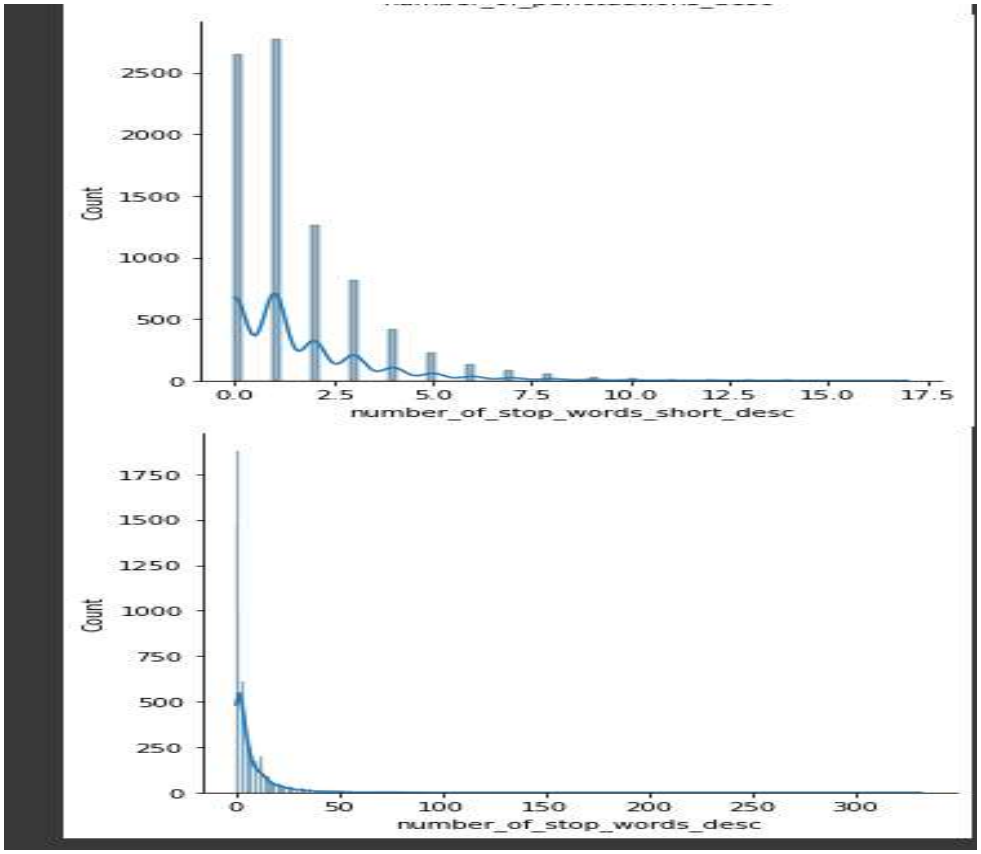
Character distribution for Short Description is also very compact



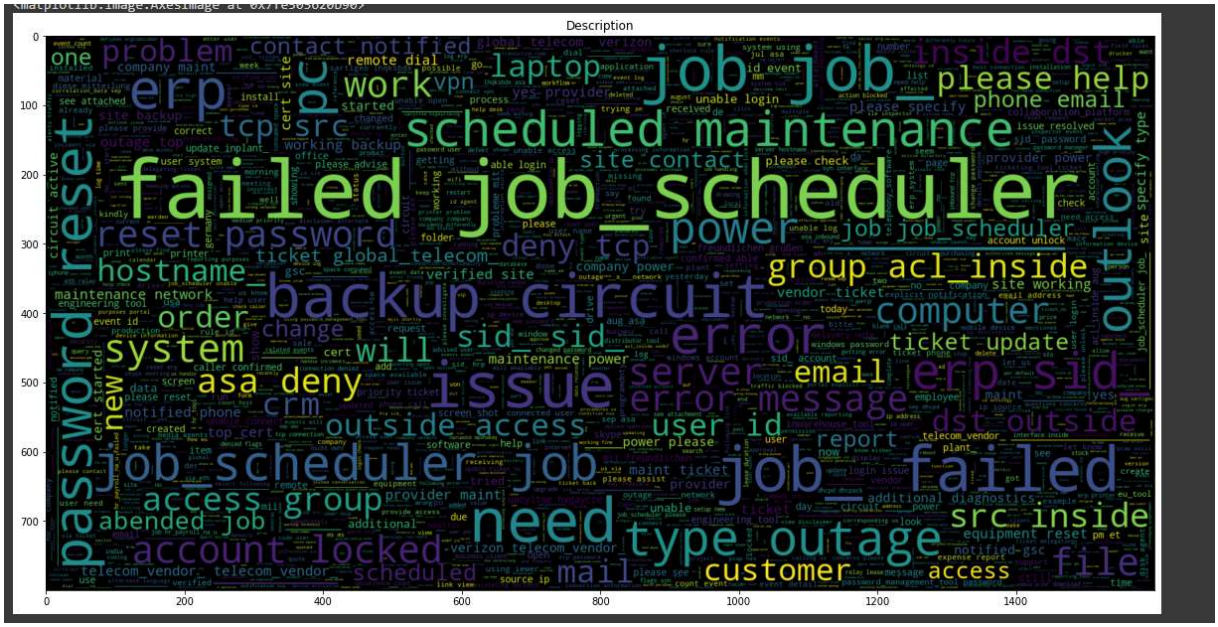
For description, in case of some tickets the character length ranges from 1000-8000 characters



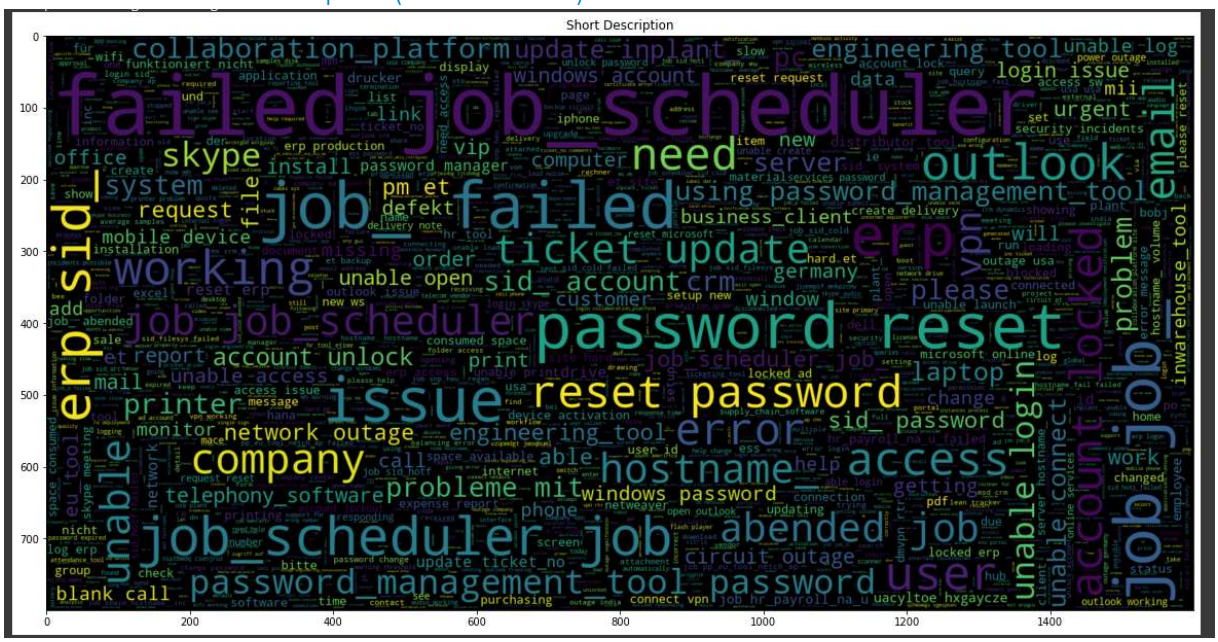
The presence of stop words is also significant in Description



Word Cloud of Description (whole data set) for max 5000 words



Word Cloud of Short Description (whole data set) for max 5000 words



Analyzing Ngram:

```
# checking bigram
(pd.Series(nltk.ngrams(ata_data['desc_analysis'], 1)).value_counts())[0:50]
```

```
(job job_ failed in job_scheduler at: // ::,)
(abended job in job_scheduler: job_ at // ::,)
(ticket update on inplant_,)
(the,)
(job hr_payroll_na_u failed in job_scheduler at: // ::,)
(password reset,)
(erp sid_ account locked,)
(windows account locked,)
(windows password reset,)
```

```
# There is pattern of text "at: // ::", these pattern suggest Time.
# We can replace these pattern with "at Time" string.
```

Replacing "at: // ::" pattern with “Time”

```
(pd.Series(nltk.ngrams(ata_data['desc_analysis'], 1)).value_counts())[0:50]
```

```
(job job_ failed in job_scheduler at Time,)
(abended job in job_scheduler: job_ at Time,)
(ticket update on inplant_,)
(the,)
(job hr_payroll_na_u failed in job_scheduler at Time,)
(password reset,)
(erp sid  account locked,)
```

Feature Engineering

Checking if any assignment group is related with any other assignment group

Finding correlated unigram and bigram between assignment group using Chi2 and TFIDF vectorization:

For GRP_0 below are most correlation unigram and bigram

```
# 'GRP_0':
  . Most correlated unigrams:
       . job_
       . failed
       . job
       . job_scheduler
  . Most correlated bigrams:
       . password reset
       . job_ failed
       . job job_
       . failed job_scheduler
```

For GRP_64, below are most correlated unigram and bigram, these are similar to GRP_0. These two group can be merged together.

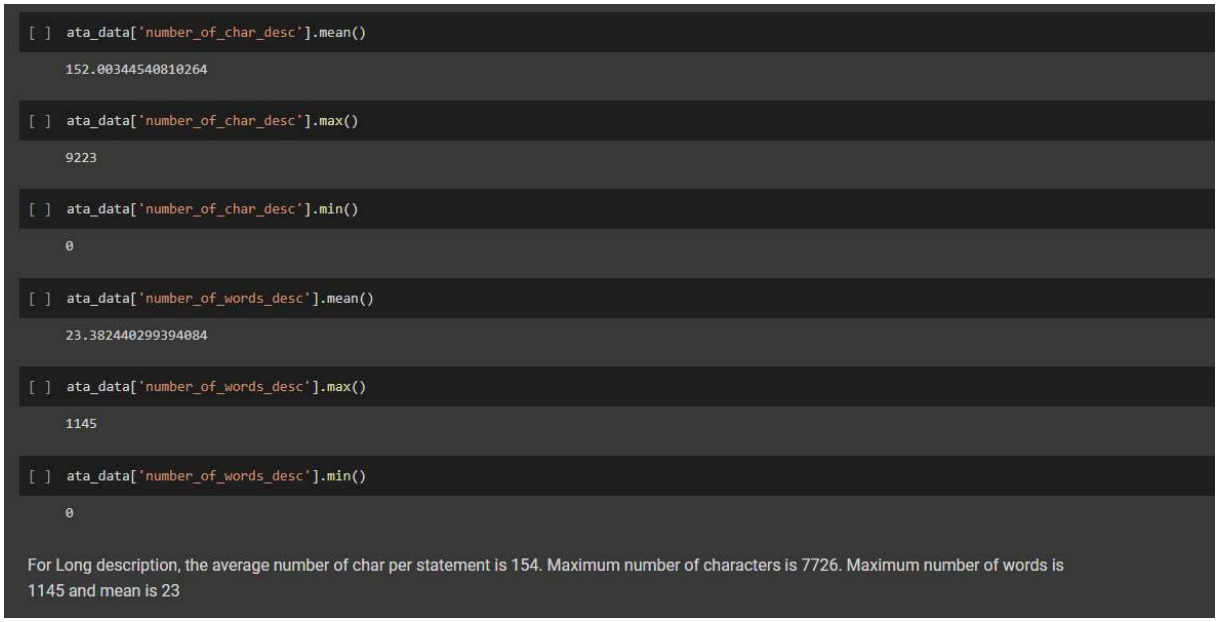
```
# 'GRP_64':
  . Most correlated unigrams:
       . waiting
       . wait
       . confirmation
       . cancel
  . Most correlated bigrams:
       . password reset
       . job_ failed
       . job job_
       . failed job_scheduler
```

After applying above analysis (on both description and short description) on whole dataset we found these assignment groups can be merged together.

1. GRP_0,GRP_35,GRP_54,GRP_58,GRP_61,GRP_64,GRP_67,GRP_70,GRP_71,GRP_17,GRP_32, GRP_38,GRP_46,GRP_49,GRP_51,GRP_52,GRP_53,GRP_54,GRP_55,GRP_58,GRP_63,GRP_66
2. GRP_1,GRP_12,GRP_47,GRP_39
3. GRP_13,GRP_29
4. GRP_10,GRP_68

Some stats on words and characters

50% of short description have 5 or less words. 75% of short description have 8 or less words



Long description indeed has lot more text as compared to short description as the mean number of words in long description is 23

Outputs from the Data-Preprocessing and EDA

This is important factor in selecting a model. We also observed that in tickets the short description gives an essence of what is happening. The long description contains lots of noise around the situation. As we need to assign the tickets to a group, we are considering short description for further analysis & model building.

Also going by the length of short description we would like to explore Bi-Directional LSTM as the algorithm. For baseline calculation, we would use Multinomial Naïve Bayes.

Handling Imbalance of dataset

RandomOverSampler is used to make the data balanced.

Algorithms used

(1) Baseline – Multinomial Naive bayes

(2) Bi-Directional LSTM

(3) Bi-Directional LSTM with CNN

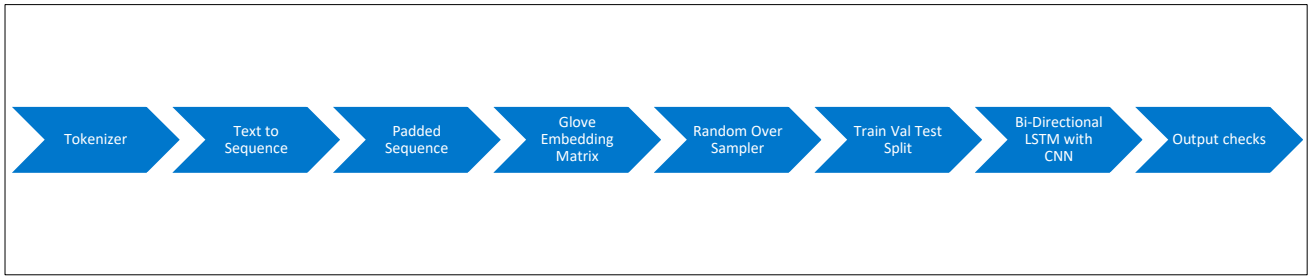
UI Creation

UI is created using flask.

Model evaluation

The final model selected is Bi-Directional LSTM with CNN. Glove embedding of 50D was used to create the embedding matrix.

Process flow for the model development



Tokenization and Text to Sequence

Getting max length of a sequence

```
[ ]  def FindMaxLength(lst):
       maxList = max((x) for x in lst)
       maxLength = max(len(x) for x in lst )
       return maxList, maxLength

[ ]  short_desc_max_list, short_desc_length_to_filter = FindMaxLength(short_desc_sequences)

[ ]  short_desc_length_to_filter

     28

[ ]  short_desc_length_to_filter_input = int(0.3*short_desc_length_to_filter)
     short_desc_length_to_filter_input

     8
```

As 75% of the short descriptions have length of 8, limiting to sentence filter to 8

Padding the sequence

The sequences are filtered to the length of 8. As 75% of the short descriptions have length of 8 or less. The filter and padding are applied as “POST” i.e. the initial 8 words are considered of the short description.

```
[ ]  def padded_sequence(sequences_input, length_to_filter_input, padding_input, truncat_input):
       padded_text_sequences = pad_sequences(sequences_input, maxlen = length_to_filter_input, padding=padding_input, truncating=truncat_input)
       return padded_text_sequences

[ ]  short_desc_padded_sequences = padded_sequence(short_desc_sequences, short_desc_length_to_filter_input, 'post', 'post')

[ ]  short_desc_padded_sequences

     array([[  18,   13,    0, ...,    0,    0,    0],
            [  16,    0,    0, ...,    0,    0,    0],
            [  28,   60,    9, ...,    0,    0,    0],
            ...,
            [ 559,   41,    7, ..., 6865, 6866,    0],
            [ 208,    5,    3, ...,    0,    0,    0],
            [ 621, 6867,   28, ...,    4,  721, 1403]], dtype=int32)

[ ]  print("Number of words:", len(short_desc_word_index)+1)

     Number of words: 6868

[ ]  short_desc_vocab_size = len(short_desc_word_index)+1
```

Inverting the indexes to re-create the short description

Vocabulary size is also determined.

Inverting the word index to recreate a short description

Inverting the indexes to re-create the short description

```
[ ]  inverted_short_desc_word_index = dict((i, word) for (word, i) in short_desc_word_index.items())
```

```
[ ]  short_desc_decoded_sequence = " ".join(inverted_short_desc_word_index[i] for i in short_desc_sequences[0])
     print(short_desc_decoded_sequence)

     login issue
```

Applying Glove embeddings

50D glove embeddings are used.

Applying Glove Word Embeddings

+ Code + Text

```
[ ]  glove_50d_file_path = '/content/drive/MyDrive/GL/NLP-1/glove.6B.50d.txt'
```

```
[ ]  short_desc_embedding_size_50d = 50
```

```
[ ]  def glove_emb_index(glove_file_path_input):
       glove_embeddings_index = {}
       glove_embedding_file = open(glove_file_path_input)

       for glove_embedding_value in glove_embedding_file:
         word_d = glove_embedding_value.split(" ")[0]
         glove_embedding = glove_embedding_value.split(" ")[1:]
         glove_embedding = np.asarray(glove_embedding, dtype='float32')
         glove_embeddings_index[word_d] = glove_embedding
       glove_embedding_file.close()
       return glove_embeddings_index
```

```
[ ]  glove_50d_embeddings_index = glove_emb_index(glove_50d_file_path)
```

```
[ ]  def glove_emb_matrix(vocab_size_input, emb_size_input, word_index_input,glove_emb_ind_input):
       glove_embedding_matrix = np.zeros((vocab_size_input, emb_size_input))
       for word, i in word_index_input.items():
         embedding_vector = glove_emb_ind_input.get(word)
         if embedding_vector is not None:
           glove_embedding_matrix[i] = embedding_vector
       return glove_embedding_matrix
```

```
[ ]  glove_50d_embedding_matrix = glove_emb_matrix(short_desc_vocab_size, short_desc_embedding_size_50d, short_desc_word_index, glove_50d_embeddings_index)
```

```
[ ]  glove_50d_embedding_matrix
```

```
     array([[ 0.        ,  0.        ,  0.        , ...,  0.        ,
              0.        ,  0.        ],
            [ 0.        ,  0.        ,  0.        , ...,  0.        ,
              0.        ,  0.        ],
            [ 0.68046999, -0.039263  ,  0.30186   , ..., -0.073297  ,
             -0.064699  , -0.26043999],
```

Label Encoding to encode the target

```
[ ]  le = LabelEncoder()
     y_ds['assignment_group'] = le.fit_transform(y_ds['assignment_group'])

[ ]  y_ds['assignment_group'].value_counts()

     0      3934
     72      645
     17      285
     4       257
     73      252
            ...
     58        1
     61        1
     68        1
     29        1
     71        1
     Name: assignment_group, Length: 74, dtype: int64

[ ]  groupings = (y_ds['assignment_group'].max()) + 1
     #groupings = len(y_ds['assignment_group'].unique())

[ ]  groupings

     74
```

Train Val Test split

```
[ ]  X_train_sd, X_val_test_sd, y_train, y_val_test = train_test_split(X_ds_sd_ros,y_ds_ros, test_size = 0.2, random_state = seed, shuffle = True, stratify = y_ds_ros)

[ ]  X_val_sd, X_test_sd, y_val, y_test = train_test_split(X_val_test_sd,y_val_test, test_size = 0.5, random_state = seed, shuffle = True, stratify = y_val_test)

[ ]  X_train_sd

     array([[ 929, 4478,   71, ..., 4479,  645,   53],
            [  34,   10,  221, ...,   19,   83,  693],
            [1821, 6190,   28, ...,    0,    0,    0],
            ...,
            [  31,   28,   14, ...,   96,   31,  156],
            [3596,    0,    0, ...,    0,    0,    0],
            [  35,   14,    2, ...,    0,    0,    0]], dtype=int32)
```

Optimizer

```
[ ]  short_desc_opt = optimizers.Adam(learning_rate=0.0001)
```

Model Summary

```
sd_glove_50d_bidi_lstm_model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 8)]               0

 embedding (Embedding)       (None, 8, 50)             343450

 spatial_dropout1d (SpatialD  (None, 8, 50)            0
 ropout1D)

 reshape (Reshape)           (None, 8, 50, 1)          0

 conv2d (Conv2D)             (None, 3, 24, 32)         320

 activation (Activation)     (None, 3, 24, 32)         0

 dropout (Dropout)           (None, 3, 24, 32)         0

 permute (Permute)           (None, 3, 32, 24)         0

 reshape_1 (Reshape)         (None, 3, 768)            0

 bidirectional (Bidirectiona  (None, 3, 256)           918528
 l)

 spatial_dropout1d_1 (Spatia  (None, 3, 256)           0
 lDropout1D)

 global_max_pooling1d (Globa  (None, 256)              0
 lMaxPooling1D)

 dense (Dense)               (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 74)                9546

=================================================================
Total params: 1,304,740
Trainable params: 1,304,740
Non-trainable params: 0
_________________________________________________________________
```

Model Compilation and Execution

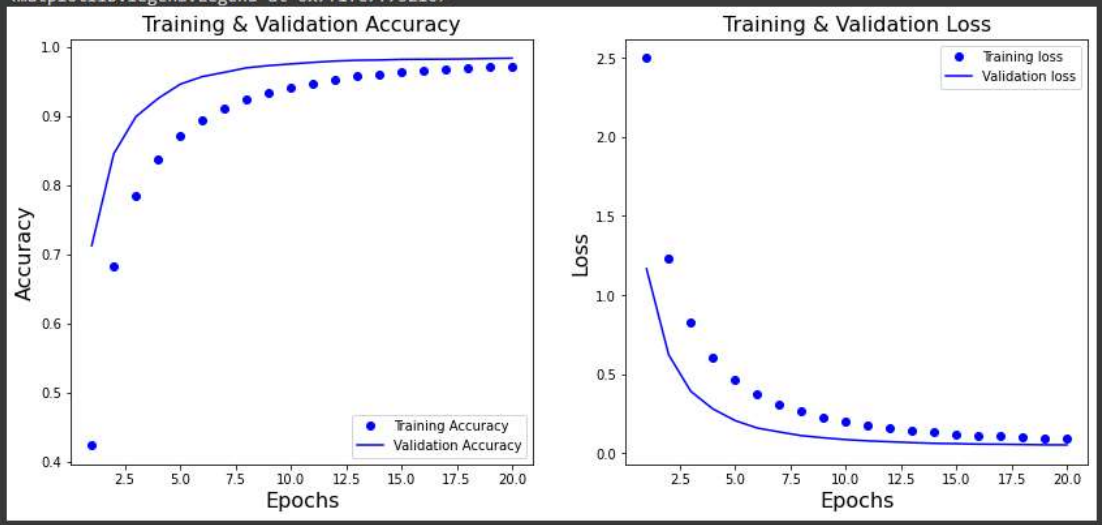
```
[ ] sd_glove_50d_bidi_lstm_model.compile(optimizer = short_desc_opt, loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])

[ ] sd_glove_50d_bidi_lstm_model_execution_data = sd_glove_50d_bidi_lstm_model.fit(X_train_sd, y_train, validation_data=(X_val_sd, y_val), epochs=20, batch_size=100, verbose=2)
```

Execution Results

```
Epoch 1/20
2329/2329 - 163s - loss: 2.4976 - accuracy: 0.4240 - val_loss: 1.1665 - val_accuracy: 0.7123 - 163s/epoch - 70ms/step
Epoch 2/20
2329/2329 - 162s - loss: 1.2343 - accuracy: 0.6827 - val_loss: 0.6229 - val_accuracy: 0.8453 - 162s/epoch - 70ms/step
Epoch 3/20
2329/2329 - 169s - loss: 0.8238 - accuracy: 0.7838 - val_loss: 0.3918 - val_accuracy: 0.8989 - 169s/epoch - 73ms/step
Epoch 4/20
2329/2329 - 171s - loss: 0.6038 - accuracy: 0.8374 - val_loss: 0.2808 - val_accuracy: 0.9251 - 171s/epoch - 74ms/step
Epoch 5/20
2329/2329 - 176s - loss: 0.4685 - accuracy: 0.8705 - val_loss: 0.2074 - val_accuracy: 0.9458 - 176s/epoch - 76ms/step
Epoch 6/20
2329/2329 - 176s - loss: 0.3755 - accuracy: 0.8944 - val_loss: 0.1601 - val_accuracy: 0.9568 - 176s/epoch - 75ms/step
Epoch 7/20
2329/2329 - 172s - loss: 0.3113 - accuracy: 0.9116 - val_loss: 0.1353 - val_accuracy: 0.9629 - 172s/epoch - 74ms/step
Epoch 8/20
2329/2329 - 168s - loss: 0.2640 - accuracy: 0.9238 - val_loss: 0.1120 - val_accuracy: 0.9695 - 168s/epoch - 72ms/step
Epoch 9/20
2329/2329 - 173s - loss: 0.2284 - accuracy: 0.9331 - val_loss: 0.0982 - val_accuracy: 0.9727 - 173s/epoch - 74ms/step
Epoch 10/20
2329/2329 - 169s - loss: 0.1995 - accuracy: 0.9412 - val_loss: 0.0867 - val_accuracy: 0.9750 - 169s/epoch - 73ms/step
Epoch 11/20
2329/2329 - 169s - loss: 0.1762 - accuracy: 0.9474 - val_loss: 0.0788 - val_accuracy: 0.9773 - 169s/epoch - 73ms/step
Epoch 12/20
2329/2329 - 172s - loss: 0.1590 - accuracy: 0.9526 - val_loss: 0.0733 - val_accuracy: 0.9793 - 172s/epoch - 74ms/step
Epoch 13/20
2329/2329 - 170s - loss: 0.1446 - accuracy: 0.9569 - val_loss: 0.0677 - val_accuracy: 0.9804 - 170s/epoch - 73ms/step
Epoch 14/20
2329/2329 - 157s - loss: 0.1318 - accuracy: 0.9606 - val_loss: 0.0631 - val_accuracy: 0.9807 - 157s/epoch - 68ms/step
Epoch 15/20
2329/2329 - 156s - loss: 0.1227 - accuracy: 0.9628 - val_loss: 0.0611 - val_accuracy: 0.9817 - 156s/epoch - 67ms/step
Epoch 16/20
2329/2329 - 161s - loss: 0.1139 - accuracy: 0.9650 - val_loss: 0.0581 - val_accuracy: 0.9819 - 161s/epoch - 69ms/step
Epoch 17/20
2329/2329 - 158s - loss: 0.1079 - accuracy: 0.9669 - val_loss: 0.0574 - val_accuracy: 0.9820 - 158s/epoch - 68ms/step
Epoch 18/20
2329/2329 - 157s - loss: 0.1001 - accuracy: 0.9694 - val_loss: 0.0556 - val_accuracy: 0.9824 - 157s/epoch - 68ms/step
Epoch 19/20
2329/2329 - 158s - loss: 0.0959 - accuracy: 0.9705 - val_loss: 0.0537 - val_accuracy: 0.9830 - 158s/epoch - 68ms/step
Epoch 20/20
2329/2329 - 159s - loss: 0.0909 - accuracy: 0.9717 - val_loss: 0.0532 - val_accuracy: 0.9834 - 159s/epoch - 68ms/step
```

Training & Validation Loss and Accuracy plots

```
<matplotlib.legend.Legend at 0x7f1fe7f73210>
```



Saving Model so that it can be used in UI

```
[ ] sd_glove_50d_bidi_lstm_model.save('short_desc_model.h5')

[ ] load_saved_model = load_model('/content/short_desc_model.h5')
```

Classification Report

```
#print(classification_report(y_true, y_pred, target_names=target_names)
print(classification_report(y_test, sd_glove_50d_bidi_lstm_model_pred.argmax(axis=1), target_names=labels))
```

```
              precision    recall  f1-score   support

      GRP_27       0.97      1.00      0.99       394
      GRP_36       0.99      0.96      0.98       393
      GRP_44       0.95      0.97      0.96       394
       GRP_2       0.96      1.00      0.98       393
       GRP_9       0.98      0.96      0.97       393
      GRP_60       1.00      1.00      1.00       393
      GRP_64       0.97      0.99      0.98       394
       GRP_8       0.94      0.98      0.96       393
      GRP_53       1.00      1.00      1.00       394
      GRP_19       0.99      1.00      0.99       394
      GRP_20       1.00      1.00      1.00       393
      GRP_69       1.00      1.00      1.00       394
       GRP_7       1.00      1.00      1.00       393
      GRP_17       0.97      0.99      0.98       393
      GRP_14       0.99      0.95      0.97       394
      GRP_72       1.00      0.97      0.98       394
      GRP_22       1.00      1.00      1.00       393
      GRP_55       1.00      1.00      1.00       394
      GRP_49       1.00      1.00      1.00       394
      GRP_65       0.94      1.00      0.97       393
      GRP_11       1.00      1.00      1.00       393
      GRP_28       0.99      0.96      0.98       393
      GRP_39       0.98      1.00      0.99       393
       GRP_1       1.00      1.00      1.00       393
      GRP_52       0.89      0.99      0.94       394
      GRP_54       0.97      1.00      0.99       394
      GRP_73       1.00      1.00      1.00       394
      GRP_61       1.00      1.00      1.00       394
      GRP_47       0.98      1.00      0.99       394
      GRP_16       1.00      1.00      1.00       393
       GRP_0       1.00      1.00      1.00       394
      GRP_48       1.00      1.00      1.00       393
      GRP_62       0.96      1.00      0.98       394
      GRP_33       1.00      1.00      1.00       394
      GRP_26       0.96      1.00      0.98       393
       GRP_4       0.95      0.97      0.96       393
      GRP_46       0.91      1.00      0.95       393
      GRP_34       1.00      1.00      1.00       393
      GRP_41       0.99      1.00      0.99       393
      GRP_56       1.00      1.00      1.00       393
      GRP_29       1.00      1.00      1.00       393
      GRP_24       0.99      1.00      1.00       393
      GRP_25       1.00      1.00      1.00       393
      GRP_32       1.00      1.00      1.00       393
      GRP_67       1.00      1.00      1.00       394
      GRP_59       1.00      1.00      1.00       393
      GRP_15       0.98      1.00      0.99       393
      GRP_45       1.00      1.00      1.00       393
      GRP_30       1.00      1.00      1.00       393
      GRP_12       0.98      1.00      0.99       394
      GRP_66       1.00      1.00      1.00       393
      GRP_70       0.98      0.77      0.86       394
      GRP_58       0.99      0.92      0.95       393

    accuracy                           0.98     29112
   macro avg       0.98      0.98      0.98     29112
weighted avg       0.98      0.98      0.98     29112
```

Checking the final classifications

```
for elem in X_test_sd:
  short_desc_decoded_sequence = " ".join(inverted_short_desc_word_index[i] for i in np.delete(elem, np.argwhere(elem == 0)))
  decoded_test_text.append(short_desc_decoded_sequence)
```

```
[ ] data_to_check = pd.DataFrame()
```

```
[ ] data_to_check['decoded_text'] = decoded_test_text
```

```
[ ] data_to_check['actual_assignment_grp'] = list(le.inverse_transform(y_test['assignment_group']))
```

```
[ ] data_to_check['pred_assignment_group'] = list(le.inverse_transform(sd_glove_50d_bidi_lstm_model_pred.argmax(axis=1)))
```

```
[ ] data_to_check.head()
```

	decoded_text	actual_assignment_grp	pred_assignment_group
0	hpqc delivers error message user is not maintained	GRP_38	GRP_38
1	need access to erp kp06789	GRP_35	GRP_35
2	make outbound calls from siemens phone system	GRP_37	GRP_37
3	help to pose the pos wh cannot be	GRP_10	GRP_10
4	tablet dell cannot connect to wireless wifi network	GRP_31	GRP_31

```
[ ] data_to_check.to_excel("output_to_check.xlsx")
```

Summary

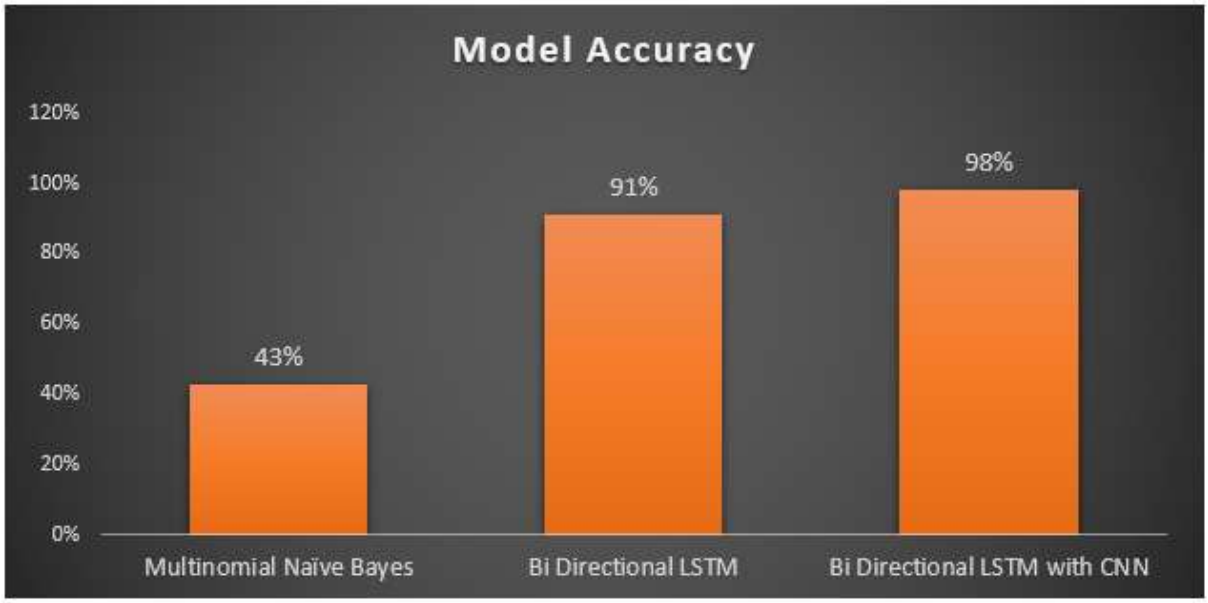
The Bi-Directional LSTM model with CNN gave an accuracy of 98% with 97% confidence

References

1. https://thesai.org/Downloads/Volume12No6/Paper_18-A_Deep_Learning_Approach_Combining_CNN_and_Bi_LSTM.pdf
2. https://arxiv.org/abs/1511.08308v5
3. https://www.researchgate.net/publication/329189864_A_Bi-Directional_LSTM-CNN_Model_with_Attention_for_Aspect-Level_Text_Classification

Comparison with Benchmark

Accuracy Comparison of different models

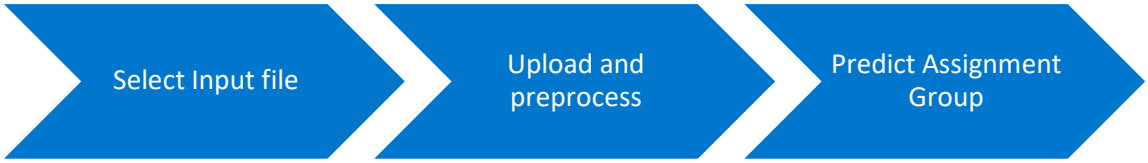


The final model using Bi Directional LSTM with CNN has 55 points improvement over the baseline accuracy Multinomial Naïve Bayes.

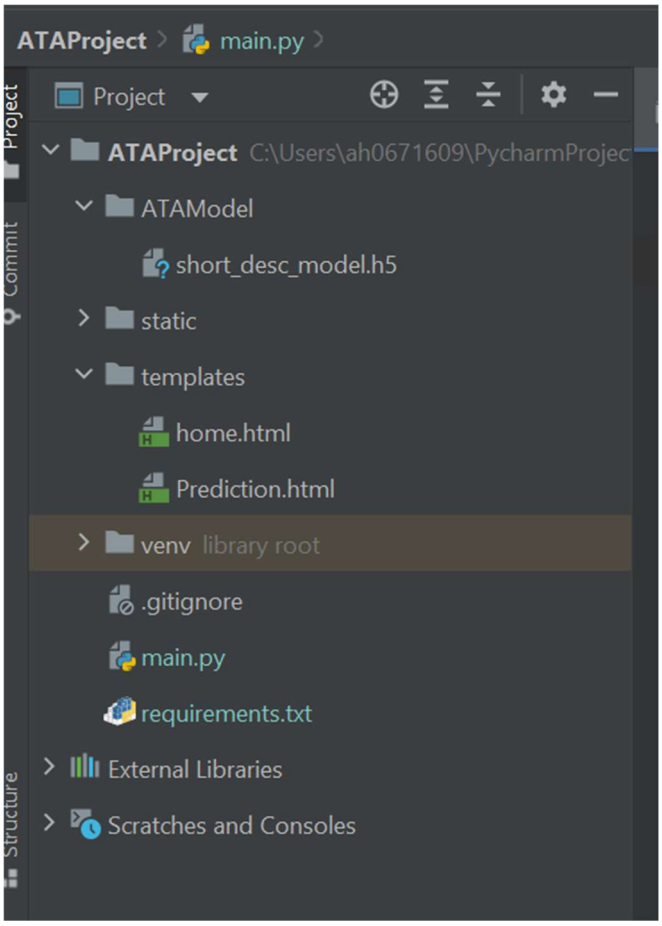
Visualizations:

Final model has integrated with Flask application for visualizing model prediction and assigning ticket to its appropriate group with highest accuracy.

Flask application workflow:

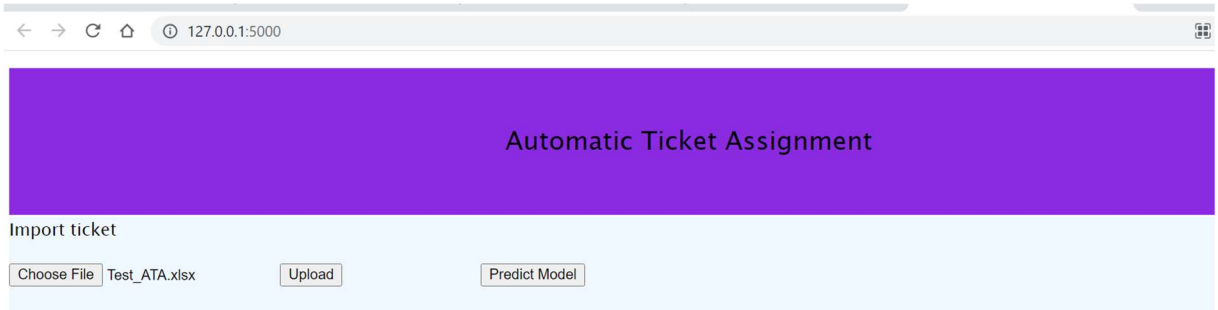


Flask application folder structure:

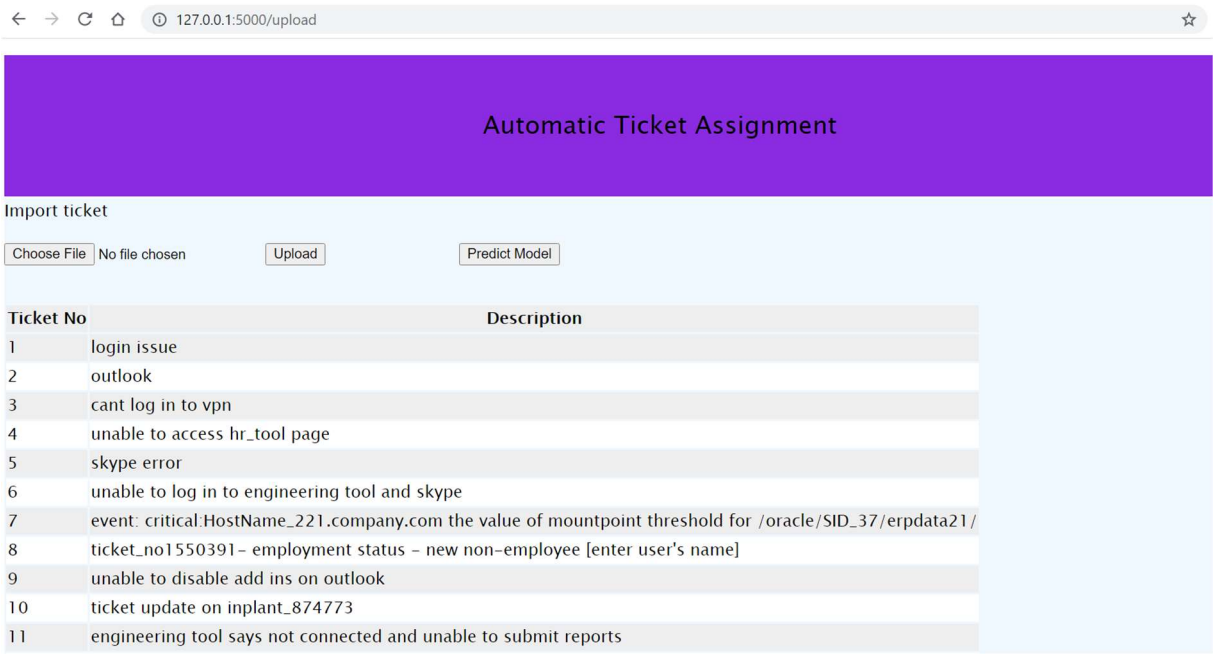


Flask application Visualization:

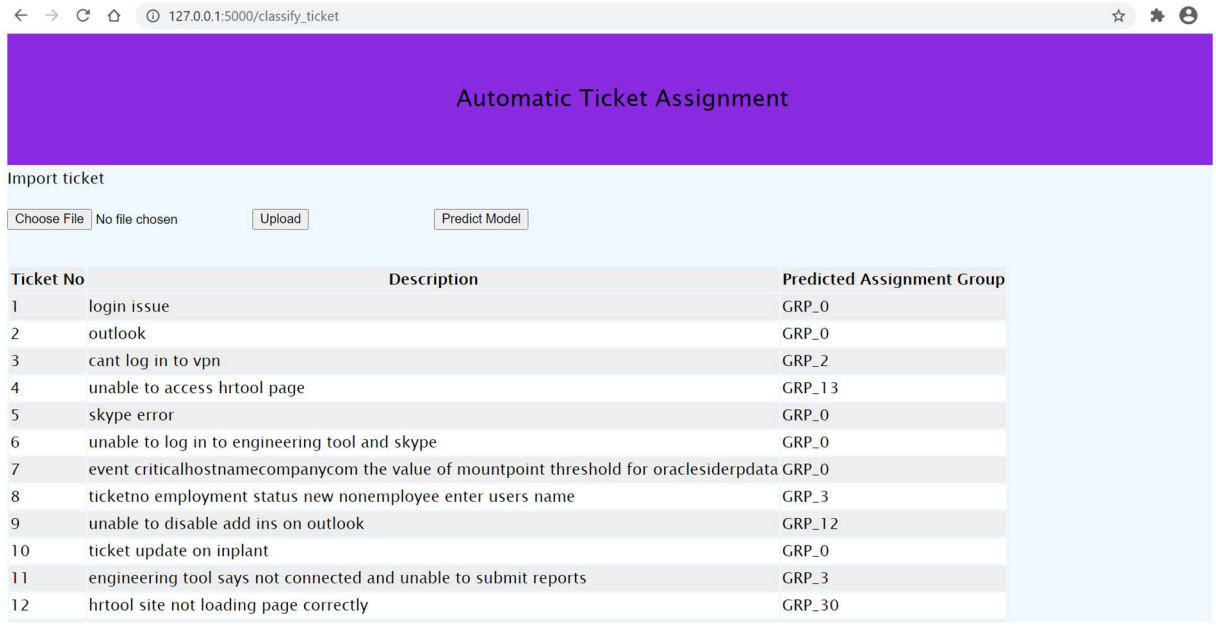
1. Select Input File:



2. Upload input file:



3. Predict Assignment Group:



Implications

The model gave an accuracy of 98% with 97% confidence.

Limitations

Data Volume

The dataset is highly imbalanced with certain groups with only 1 entry. It would be helpful if the volume of the data is more for such groups.

Data Quality

There are below scenarios where a particular ticket, in the below example job_593 was assigned to GRP-8 26 times, but was assigned to GRP_5 only 1 time. It can be a case of misclassification. We have currently not done any correction in data for such scenarios. Excluding/correcting such cases would further increase the model performance.

11	GRP_0	30
12	**job_593**	**27**
13	GRP_5	1
14	GRP_8	26
15	**erp sid_34 password reset**	**23**
16	GRP_0	23

Closing Reflections

We found the data was present in multiple languages and in various formats such as emails, chat, automated mails, etc bringing in a lot of variability in the data to be analyzed. The Business can improve the process of raising tickets via a common unified IT Ticket Service Portal which reduces the above mentioned variability. By doing this, the model can perform better which can help businesses to identify the problem area for relevant clusters of topics.

Also reducing the number of assignment group would help in easing the complexity of the overall system.