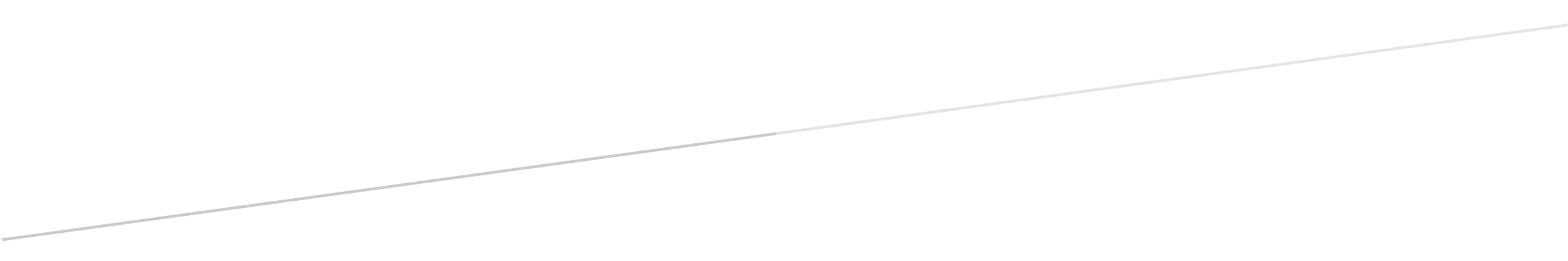
Automatic Ticket Assignment

Assigning ticket to appropriate group

Asjad, Kamal, Lakshmi, Pankaj



Contents

[Summary of problem statement, data, and findings 3](#_Toc90197057)

[Other business use case of text classification 4](#_Toc90197058)

Overview of the final process[: Pre-processing 4](#_Toc90197059)

[Data Shape 4](#_Toc90197060)

[Some more info on the dataset 4](#_Toc90197061)

[Quick peek into data 4](#_Toc90197062)

[Check for nulls 5](#_Toc90197063)

[Handling Nulls 5](#_Toc90197064)

[Identifying Duplicates 6](#_Toc90197065)

[Removing duplicates 6](#_Toc90197066)

[Mojibake 6](#_Toc90197067)

[Presence of non-English language 6](#_Toc90197068)

[Checking for Patterns: "received from: *eylqgodm.ybqkwiam@gmail.com*" 7](#_Toc90197069)

[Checking for Patterns: "email ids" 7](#_Toc90197070)

[Checking for Patterns: "Mail Format" 8](#_Toc90197071)

[Checking for Patterns: <mailto:> 9](#_Toc90197072)

[Checking for Patterns: template with name, language, browser, etc. 9](#_Toc90197073)

[Checking for Patterns: Checking for embedded images text 10](#_Toc90197074)

[Checking for Patterns: Phrases 11](#_Toc90197075)

[Checking for Patterns: Expanding acronyms such as “pls” and replacing phrase such as “please help to” 11](#_Toc90197076)

[Expanding contractions 12](#_Toc90197077)

[Overview of the final process: EDA 12](#_Toc90197078)

[Distribution of tickets as per Assignment Group 12](#_Toc90197079)

[Words/Characters analysis 13](#_Toc90197080)

[Word Cloud of Description (whole data set) for max 5000 words 16](#_Toc90197081)

[Word Cloud of Short Description (whole data set) for max 5000 words 16](#_Toc90197082)

[Analyzing Ngram: 16](#_Toc90197083)

[Replacing "at: // ::" pattern with “Time” 17](#_Toc90197084)

[Feature Engineering 17](#_Toc90197085)

Model evaluation  [1](#_Toc90197078)8

Comparison to benchmark  [1](#_Toc90197078)9

Visualizations 20

Implications 21

Limitations 22

Closing Reflection23

# 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.

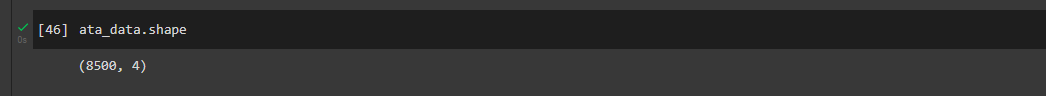
## Other business use case of text classification

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

# Overview of the final process: Pre-processing

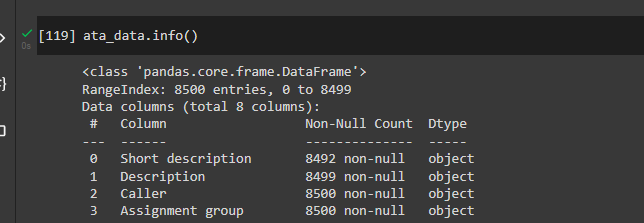
## Data Shape

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

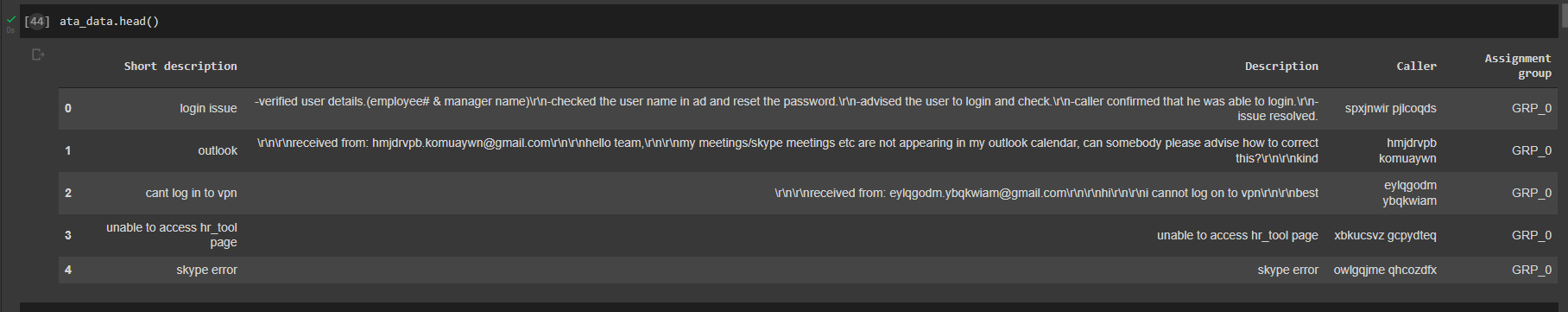


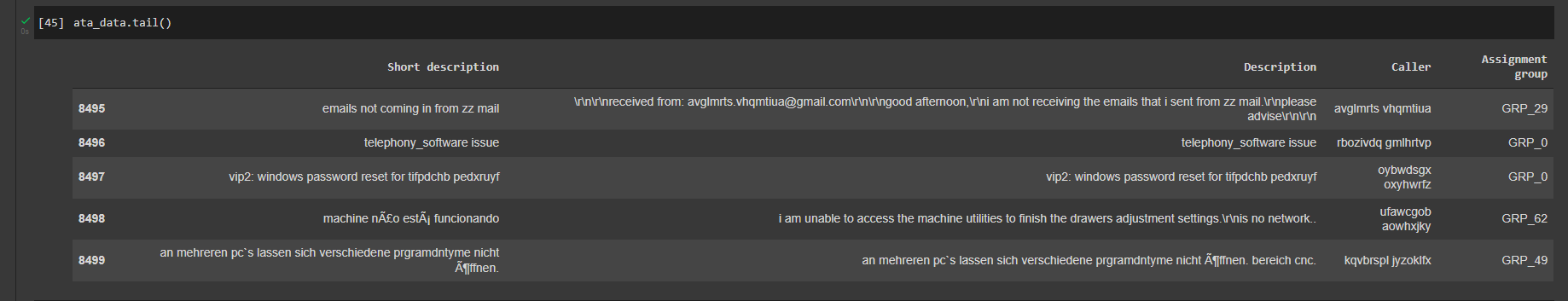
## Some more info on the dataset

The four columns are:-



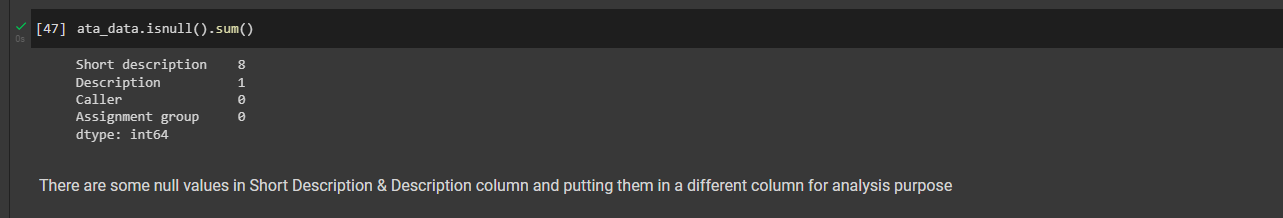
## Quick peek into data

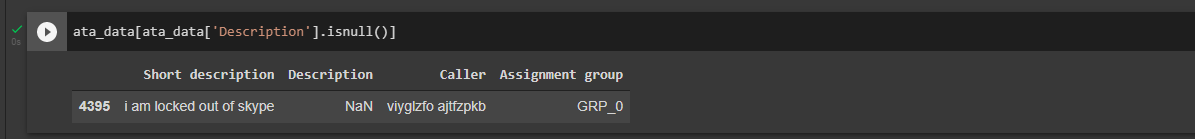


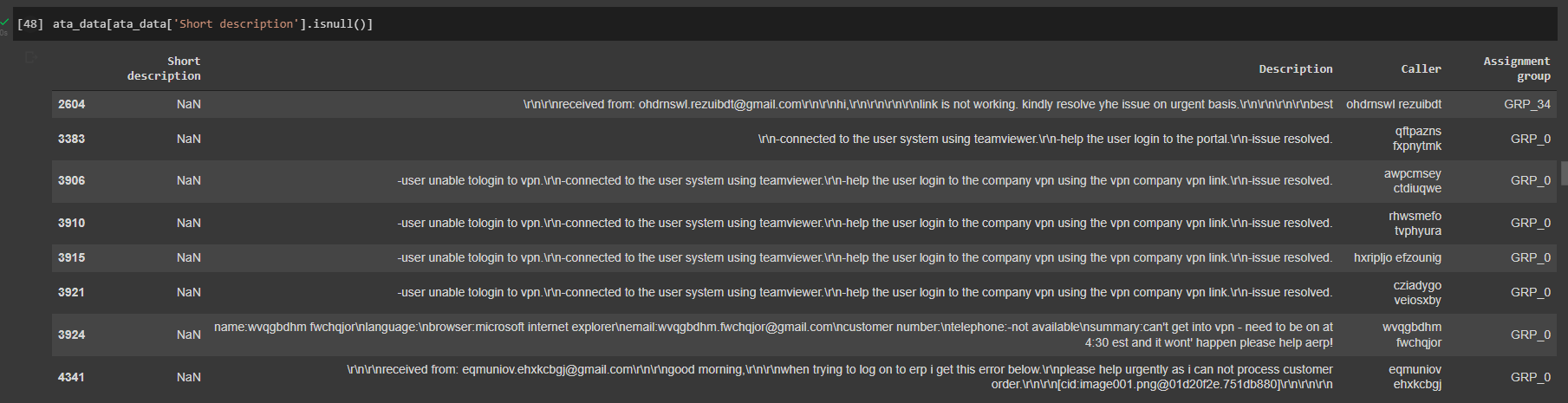


## Check for nulls

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

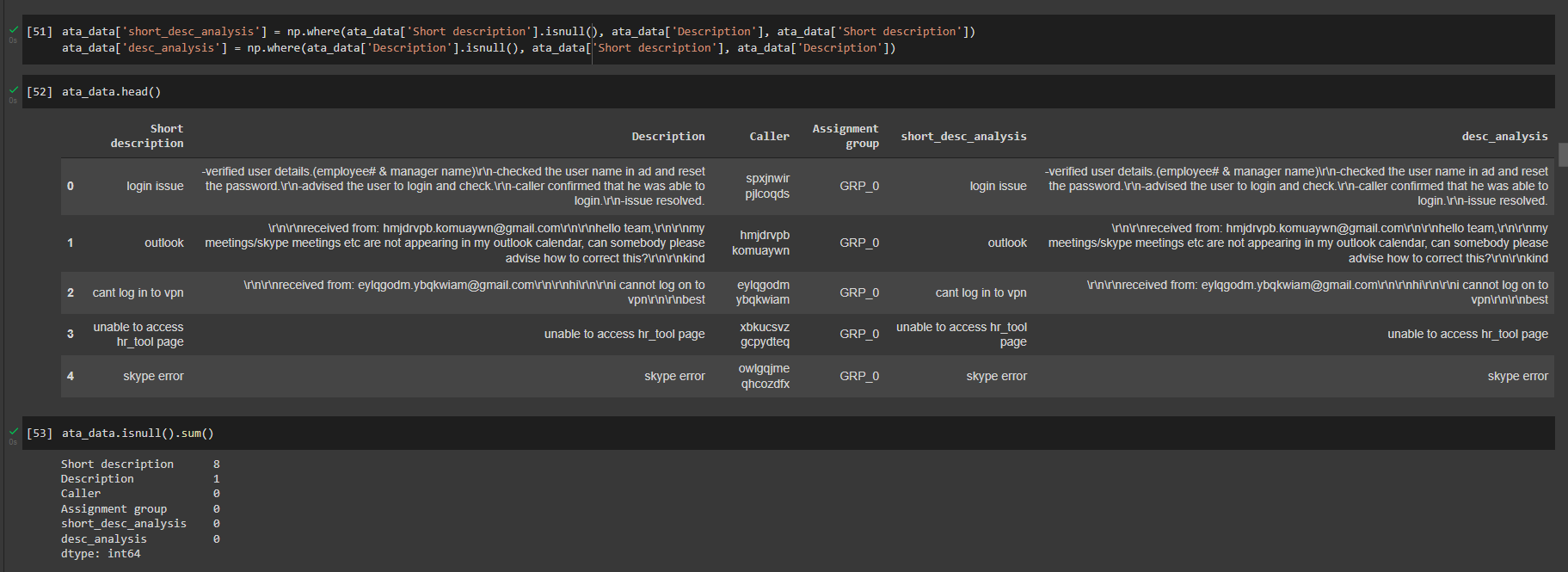




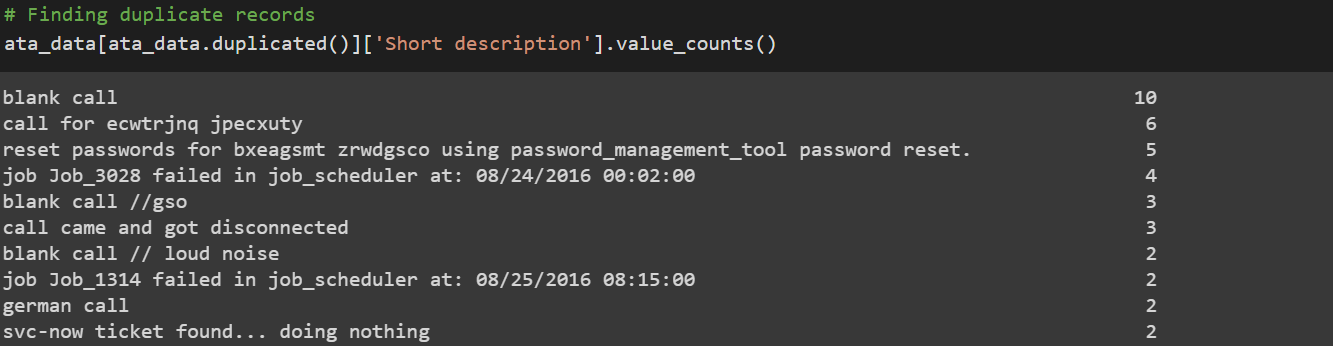


## Handling Nulls

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

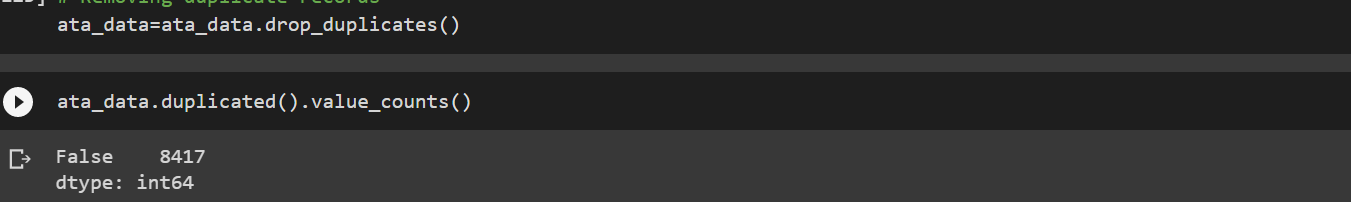


## Identifying Duplicates



## Removing duplicates

After removing duplicates remaining records: 8417



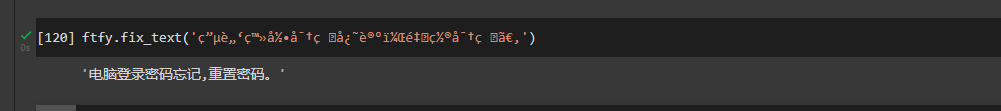
## Mojibake

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



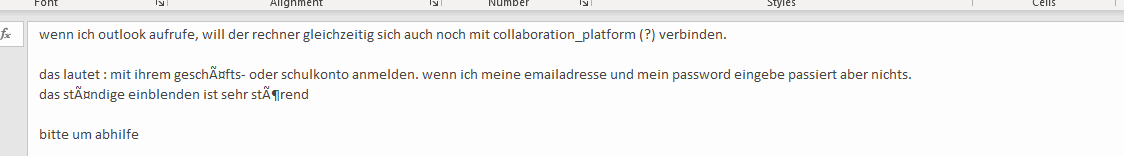
Reference: <https://www.kaggle.com/rtatman/data-cleaning-challenge-character-encodings>

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

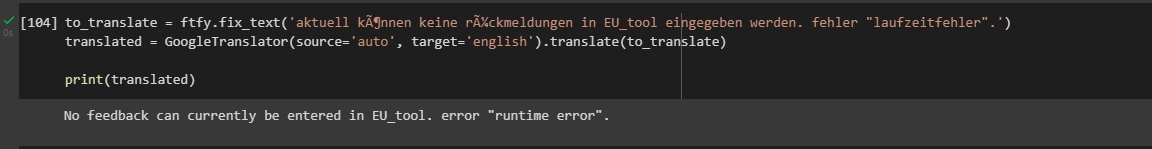


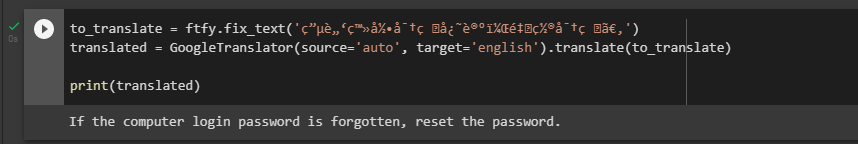
## Presence of non-English language

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



**GoogleTranslator** is used to translate non-English text to English

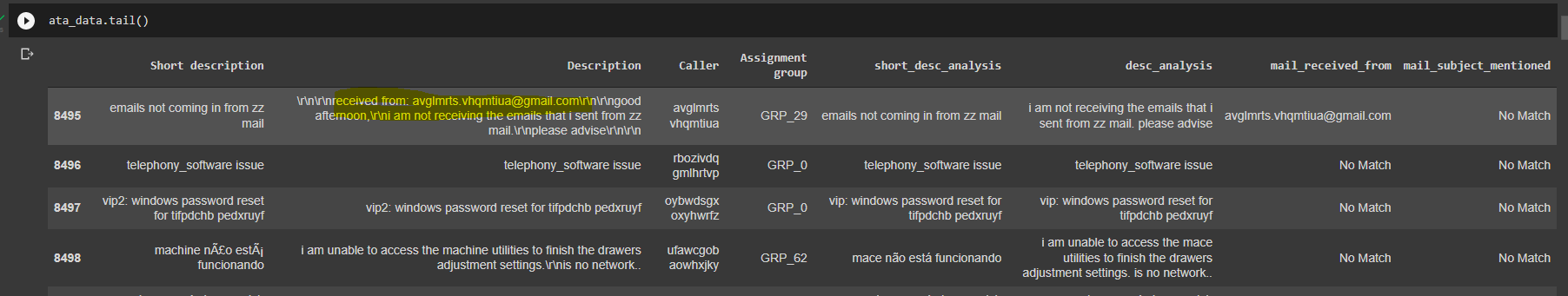




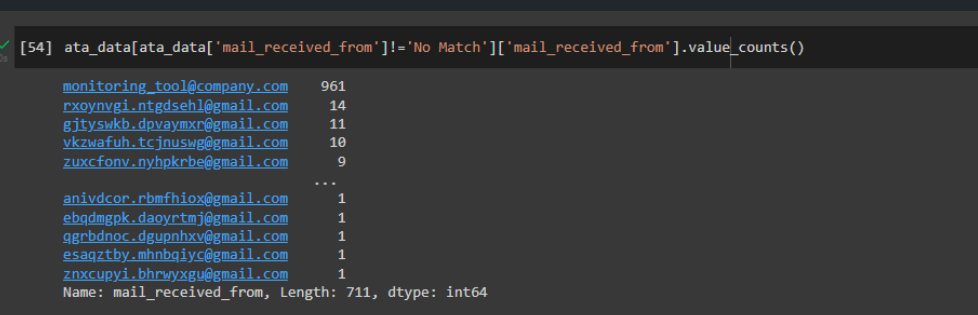
## Checking for Patterns: "received from: [*eylqgodm.ybqkwiam@gmail.com*](mailto:eylqgodm.ybqkwiam@gmail.com)"

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

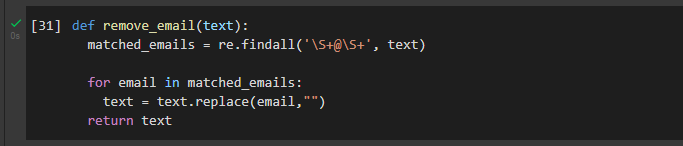


Analyzing the mails received from indicates that there are 961 records where initially issue was triggered due to system generated mail from monitoring\_tool@company.com



## Checking for Patterns: "email ids"

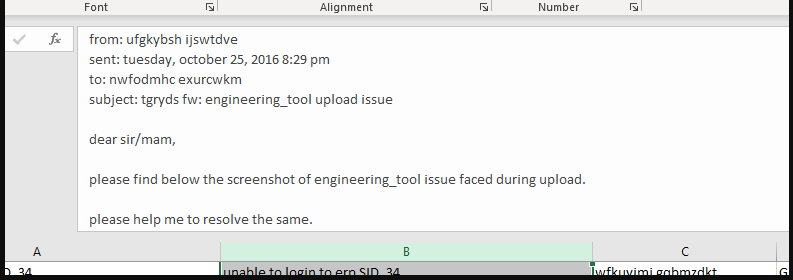
A description could contain multiple email ids. Removing that pattern and replacing with blank



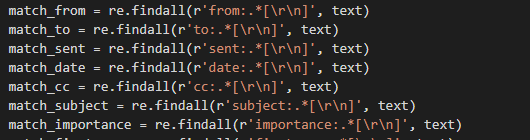
## Checking for Patterns: "Mail Format"

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



Clean text



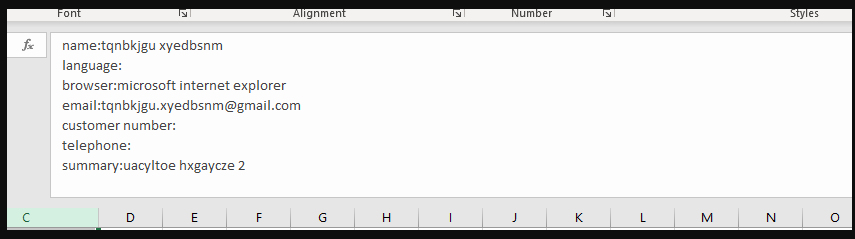
## Checking for Patterns: <mailto:>

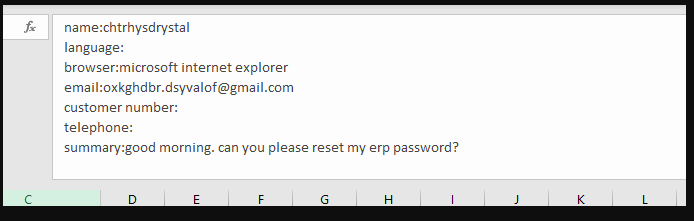
Removing occurrences of mailto: from the text



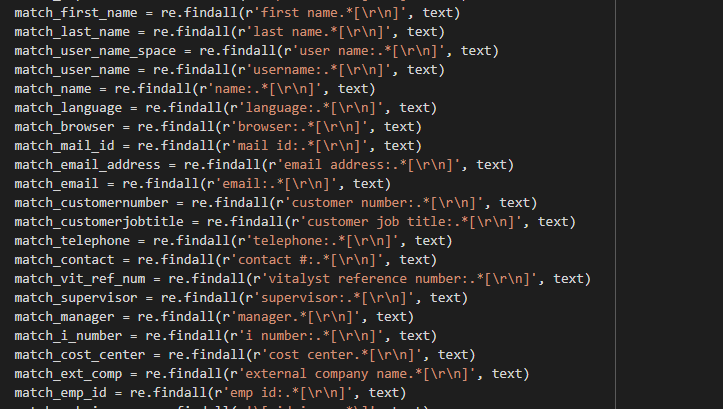
## Checking for Patterns: template with name, language, browser, etc.

The description column has a template for e.g.





Removing such pattern using regex



## Checking for Patterns: Checking for embedded images text

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



Regex is used to replace such patterns with blanks

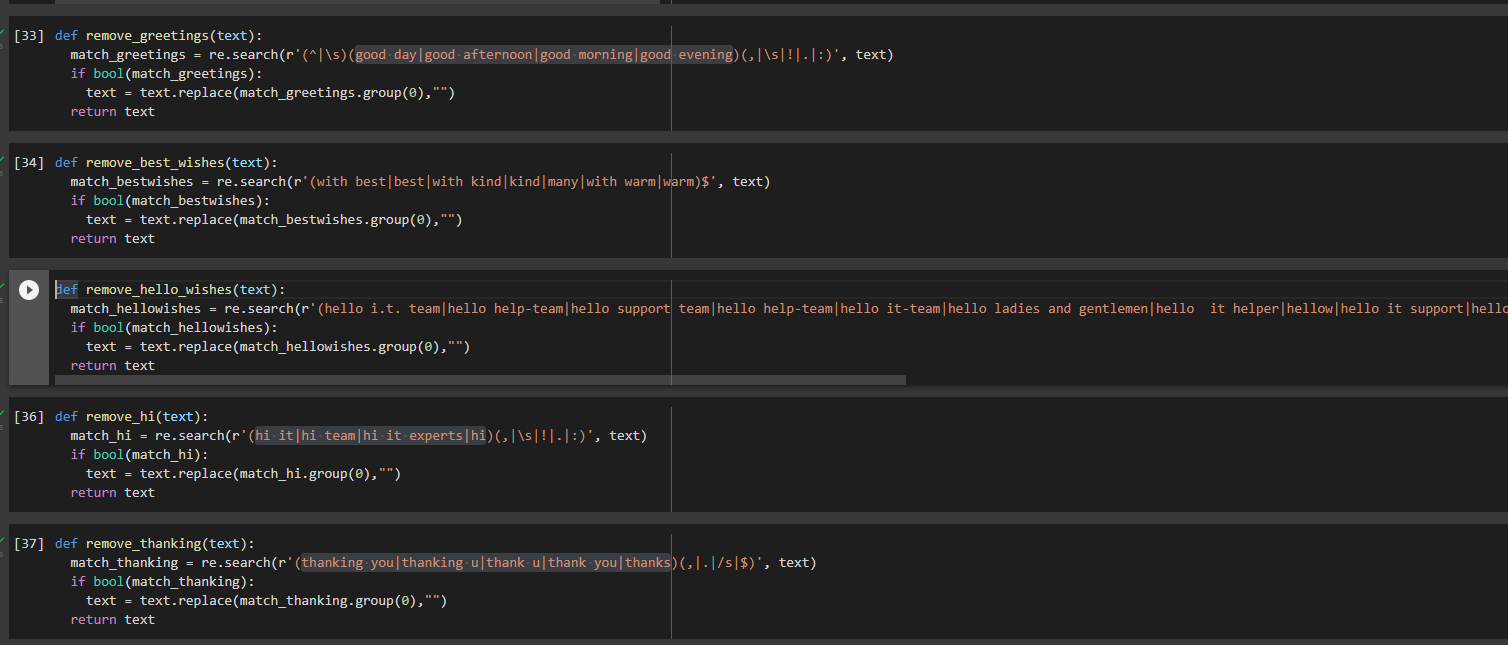


## Checking for Patterns: Phrases

Certain pattern of text found in corpus

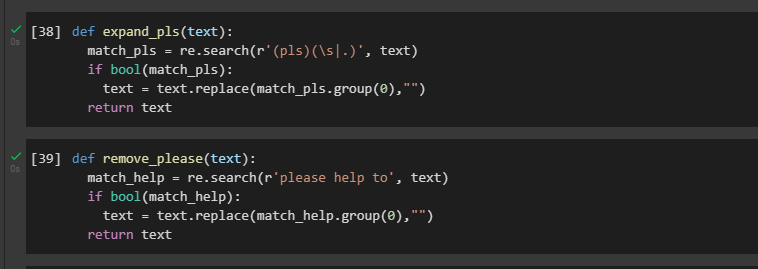
* begin forwarded message:
* sent from my iphone
* sent from my ipad
* “sir or madam,” or “sir/mam,” or “sir,”
* yes/no/na
* good day or good afternoon or good morning or good evening
* hello i.t. team or hello help-team or hello support team or hello help-team or hello it-team or hello ladies and gentlemen 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



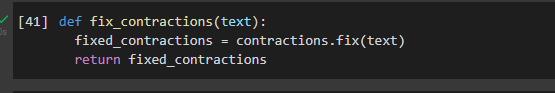
## Checking for Patterns: Expanding acronyms such as “pls” and replacing phrase such as “please help to”

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

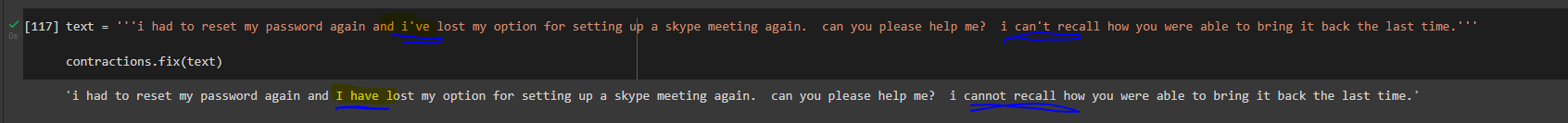


## Expanding contractions

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**

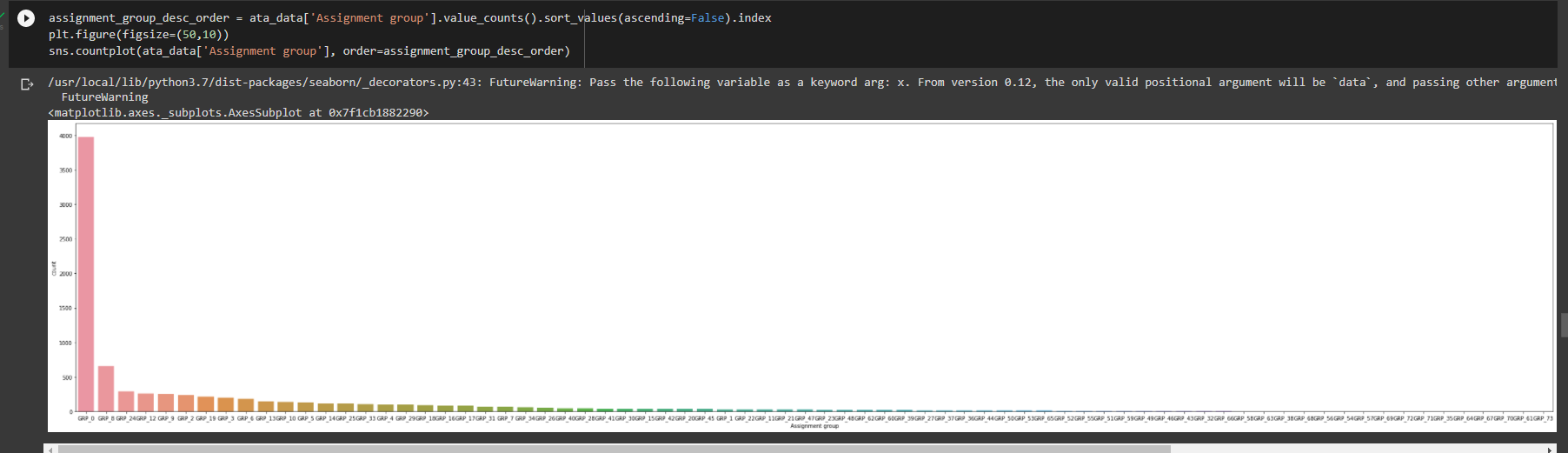


Applying contractions.fix



# Overview of the final process: EDA

## Distribution of tickets as per Assignment Group

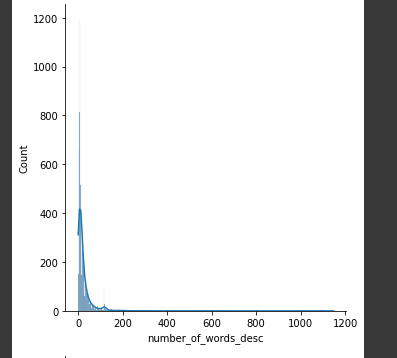


Group\_0 has 3976 tickets (roughly 47% of tickets). Dataset is imbalanced

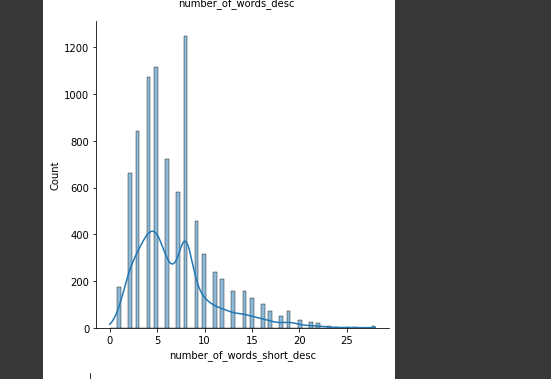
* no. of group less than 10 tickets- 25
* no. of groups in which records between 10 and 100 - 37
* no. of group greater than 100 tickets – 11

## Words/Characters analysis

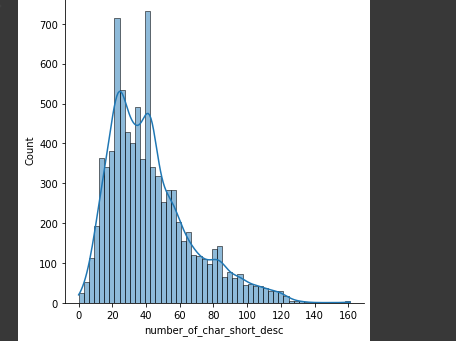
Majority of tickets have upto 100 words in Description. However, there are some tickets ranging from 200 – 1200 words



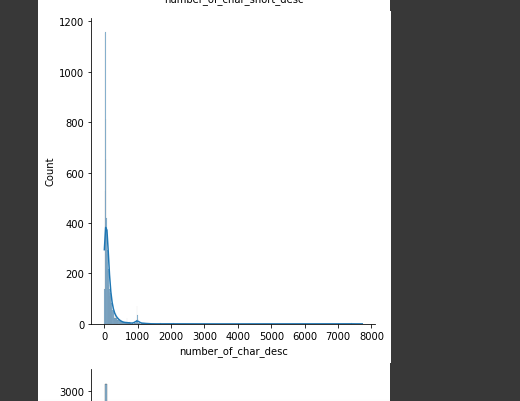
Short Description has a much compact word distribution



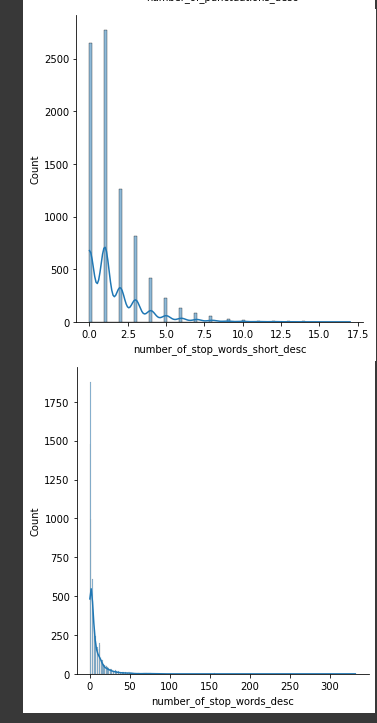
Character distribution for Short Description is also very compact



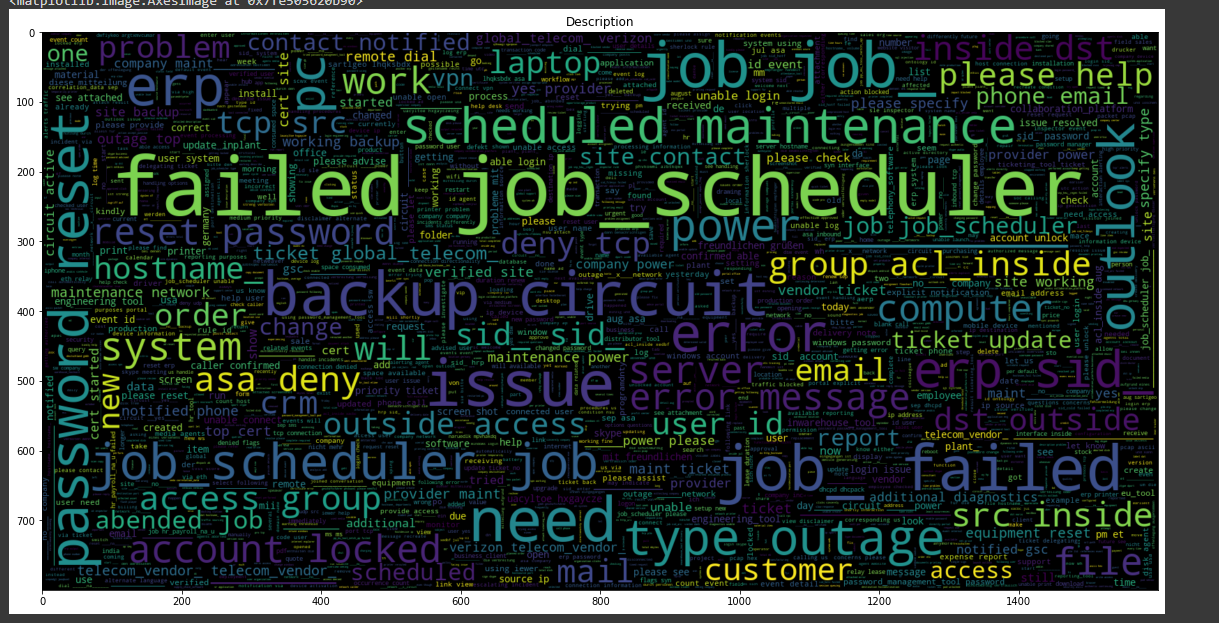
For description, in case of some tickets the character length ranges from 1000-8000 characters



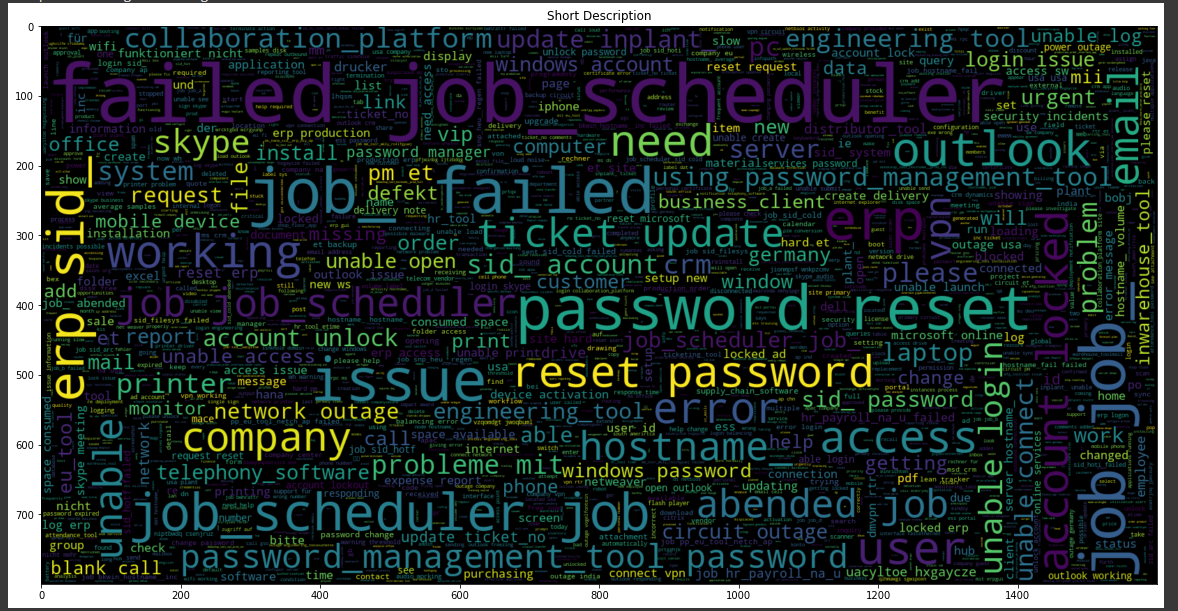
The presence of stop words is also significant in Description



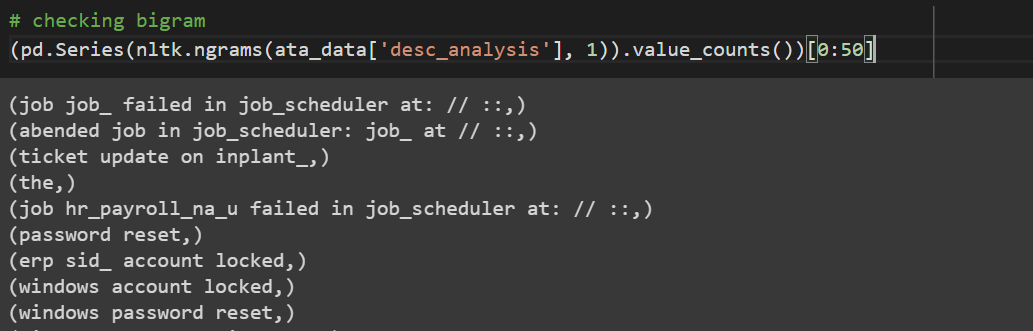
## Word Cloud of Description (whole data set) for max 5000 words



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



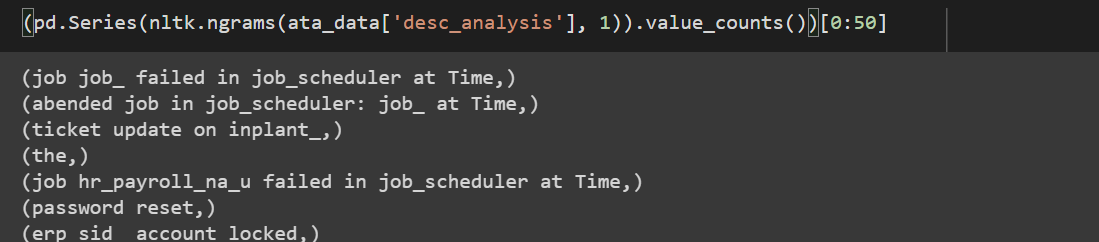
## Analyzing Ngram:



# There is pattern of text "at: // ::", these pattern suggest Time.

# We can replace these pattern with "at Time" string.

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



## 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



For GRP\_64, below are most correlated unigram and bigram, these are similar to GRP\_0. These two group can be merged together.



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

# Comparison to benchmark

From the given problem description, we could see that the existing system is able to assign XYZ% of the tickets correctly.

So our objective here is to build an AI-based classifier model to assign the tickets to right functional groups by analysing the given description with an accuracy of at least XYZ%.

From the prediction results we see that the GRU model based on the resampled data is able to achieve an accuracy of XYZ% which is above our benchmark.

**Implications**

Although this model can classify the IT tickets with XYZ% accuracy, to achieve better accuracy in the real world it would be good if the business can collect additional data around 300 records for each group.

**Limitations**

As part of Data pre-processing, we had grouped all assignment groups with less than 10 entries as one group (misc\_grp) which had reduced the Target class from 74 to 50 groups. While applying this model in the real world there could be additional intervention required to classify the tickets if it has been classified as misc\_grp by our model. Since the number of elements reported under misc\_grp are less, we expect this intervention to be done less often.

**Closing Reflections**

We found the data was present in multiple languages and in various formats such as emails, chat, 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.