Automatic Ticket Assignment

Assigning ticket to appropriate group

Contents

[The Real Problem 2](#_Toc90197055)

[Business Domain Value 3](#_Toc90197056)

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

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

[Summary of the Approach to EDA and Pre-processing: 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)

[Summary of the Approach to EDA and Pre-processing: 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)

[Deciding Models and Model Building 18](#_Toc90197086)

[Rule based model 18](#_Toc90197087)

[ML based model 18](#_Toc90197088)

[Traditional ML Model 19](#_Toc90197089)

[Neural Network based Model 20](#_Toc90197090)

[Summary 22](#_Toc90197091)

[How to improve your model performance? 22](#_Toc90197092)

[Use of embeddings 22](#_Toc90197093)

[Reduction of noise 22](#_Toc90197094)

[Use of Correlated unigram and bigram between assignment group 22](#_Toc90197095)

# 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. ©Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited 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

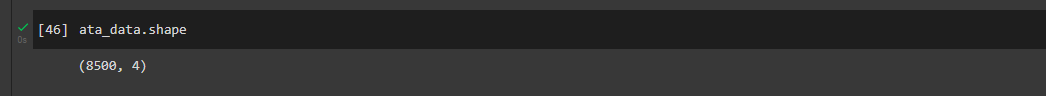
Automatic Ticket Assignment (ATA) is a classification problem which comes under the Supervised Machine Learning category & plays a key role for successfully running any Incident Management System, especially in very large system that provides numerous services, and each service has multiple categories 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

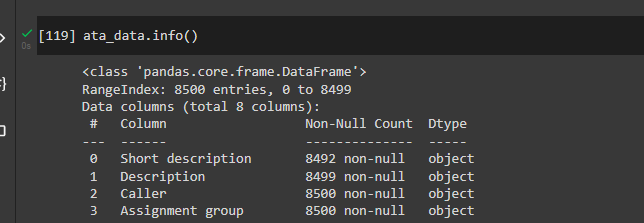
## Data Shape

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

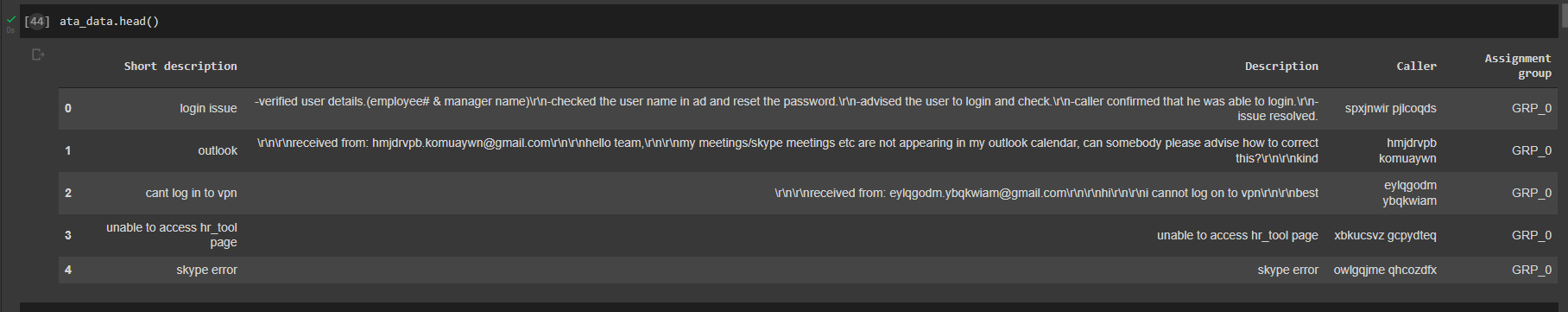


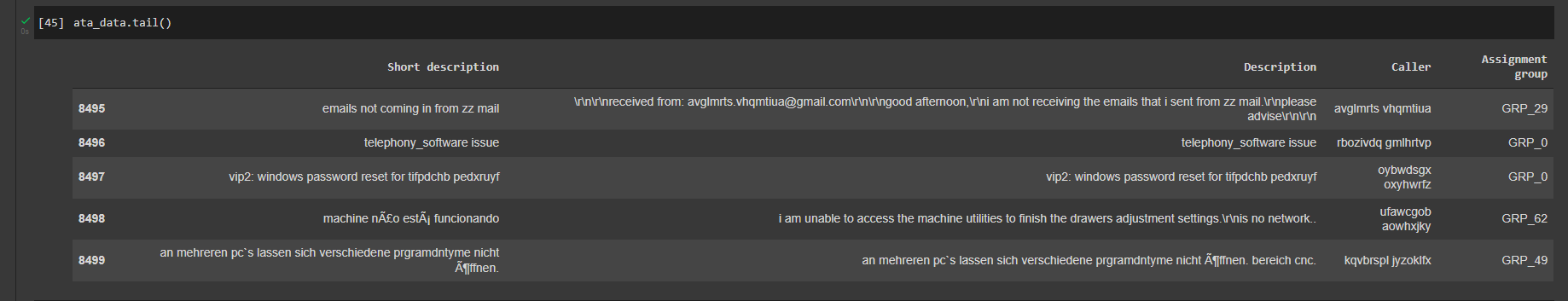
## Some more info on the dataset

The four columns are:-



## Quick peek into data





## 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

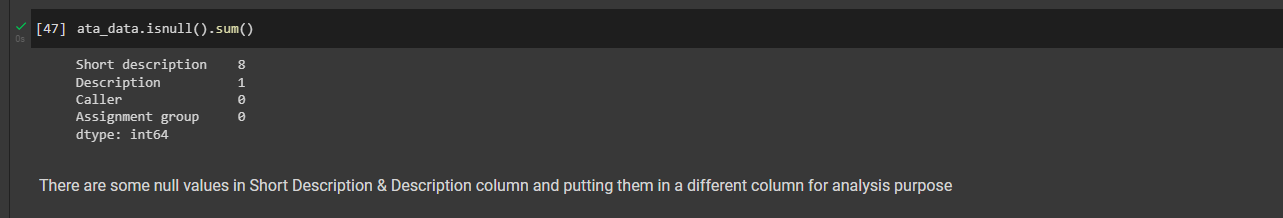
## Data pre-processing steps

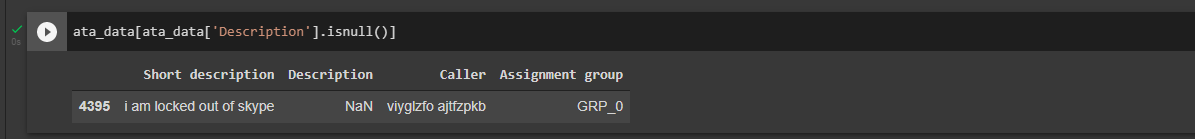
# Step by Step Walkthrough the solution

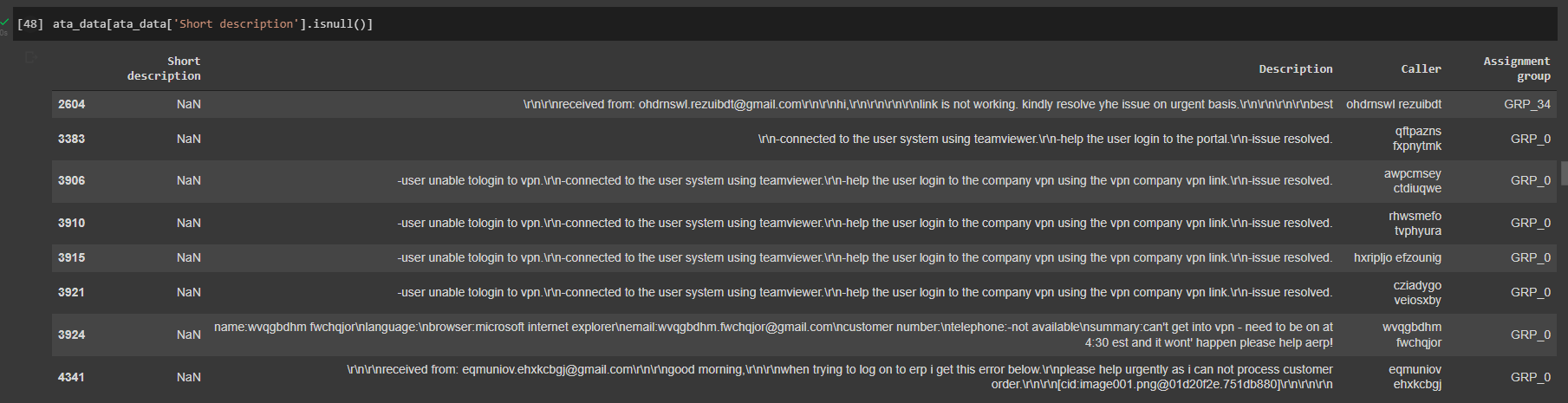
## **Challenges faced in cleaning data**

## Check for nulls

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

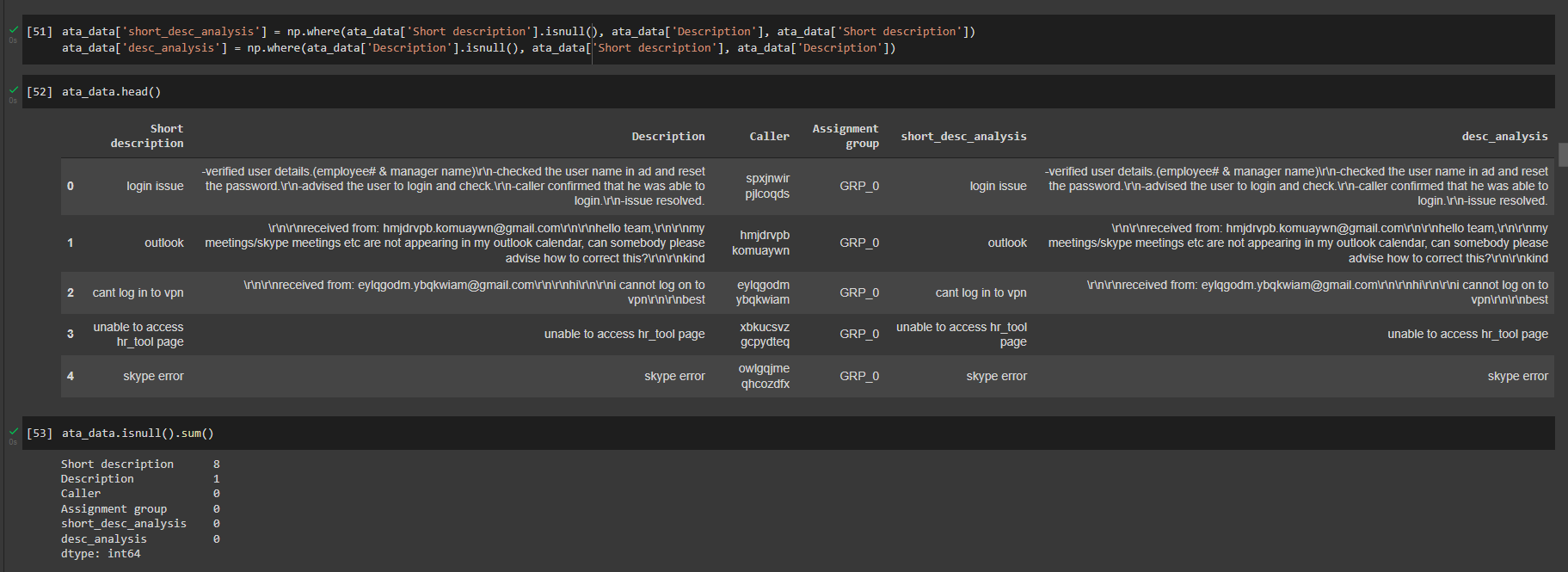




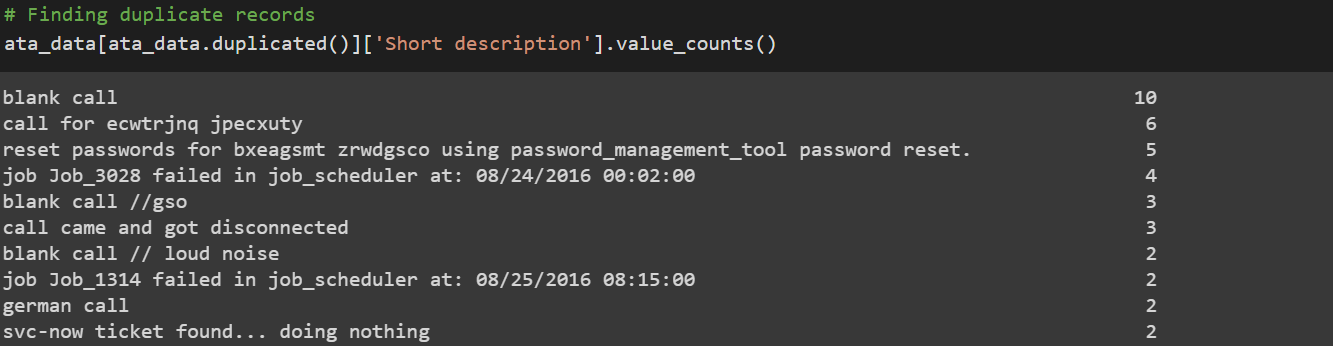


## Handling Nulls

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

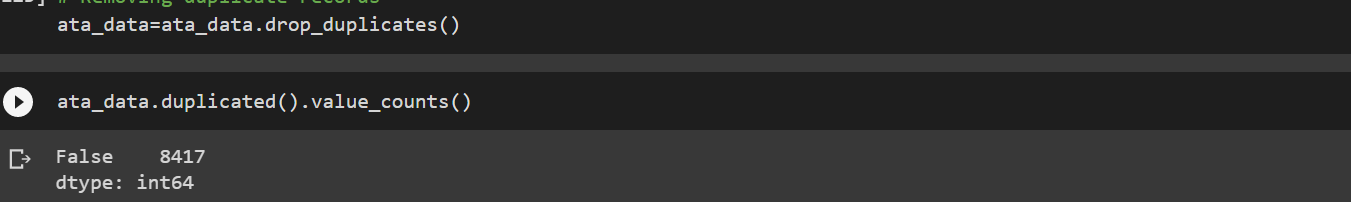


## Identifying Duplicates



## Removing duplicates

After removing duplicates remaining records: 8417



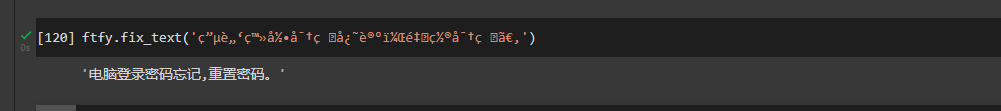
## Mojibake

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



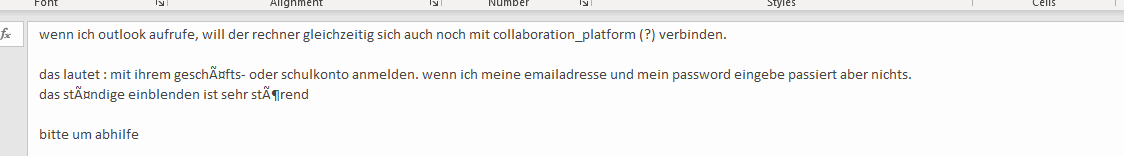
Reference: <https://www.kaggle.com/rtatman/data-cleaning-challenge-character-encodings>

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

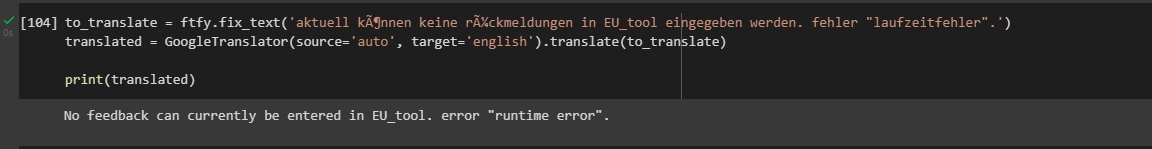


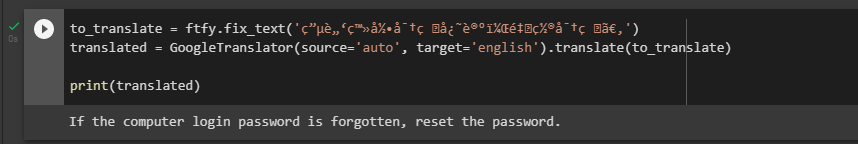
## Presence of non-English language

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

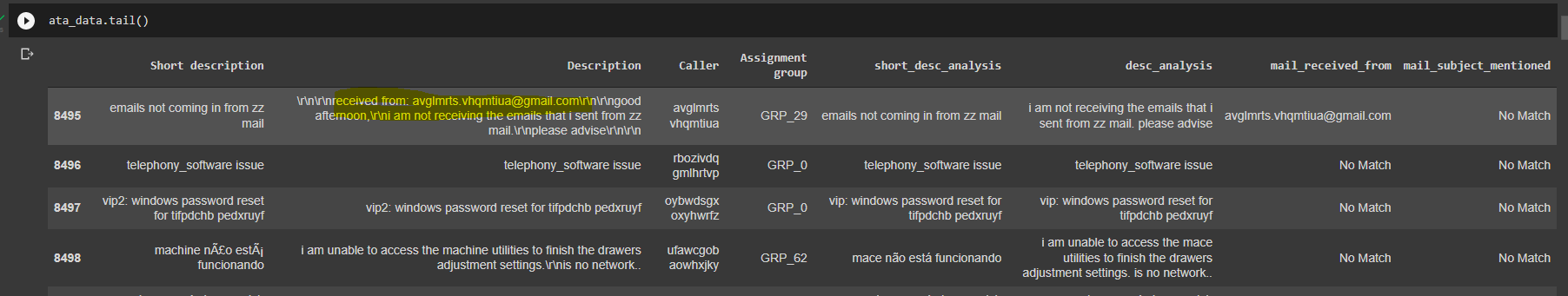




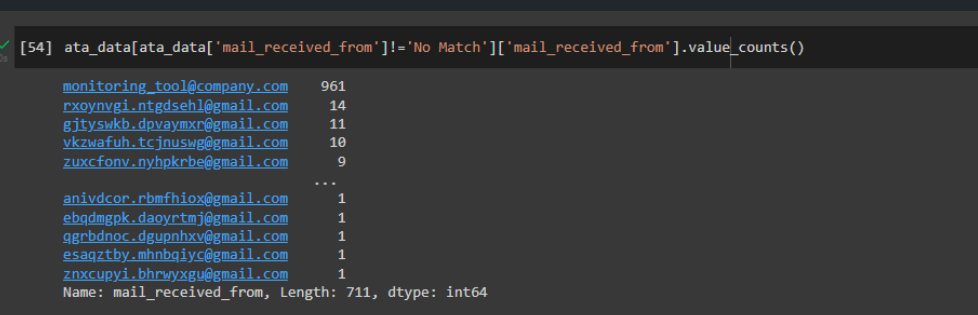
## Checking for Patterns: "received from: [*eylqgodm.ybqkwiam@gmail.com*](mailto:eylqgodm.ybqkwiam@gmail.com)"

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

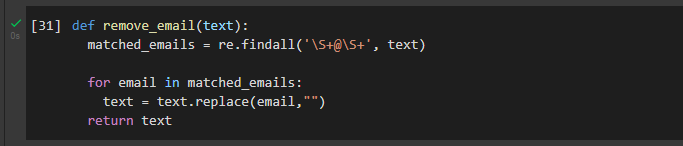


Analyzing the mails received from indicates that there are 961 records where initially issue was triggered due to system generated mail from monitoring\_tool@company.com



## Checking for Patterns: "email ids"

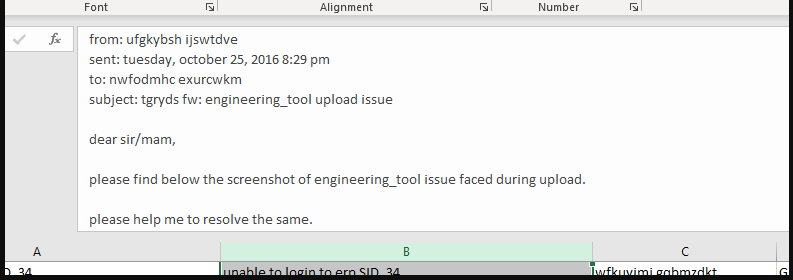
A description could contain multiple email ids. Removing that pattern and replacing with blank



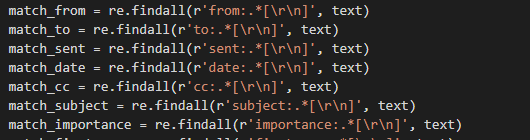
## Checking for Patterns: "Mail Format"

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



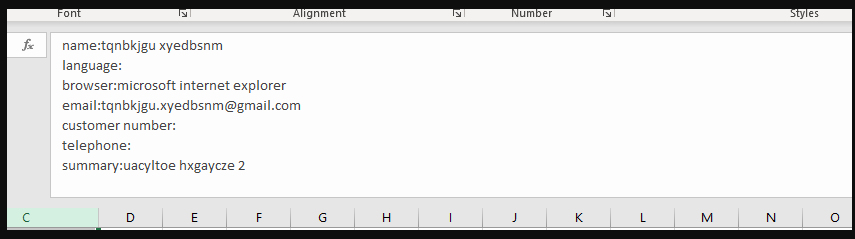
## Checking for Patterns: <mailto:>

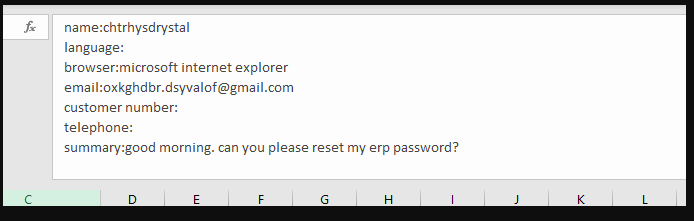
Removing occurrences of mailto: from the text



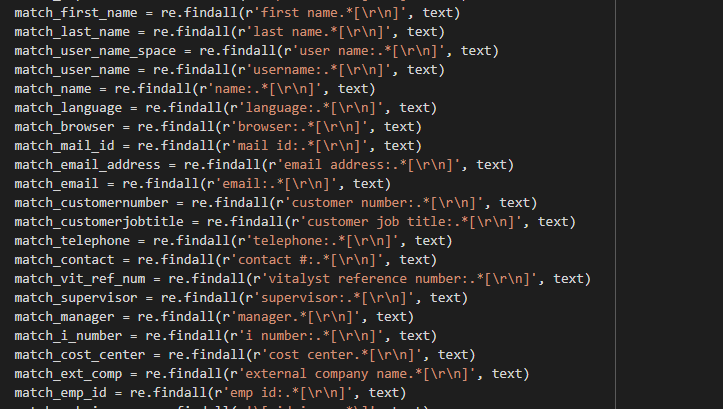
## Checking for Patterns: template with name, language, browser, etc.

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

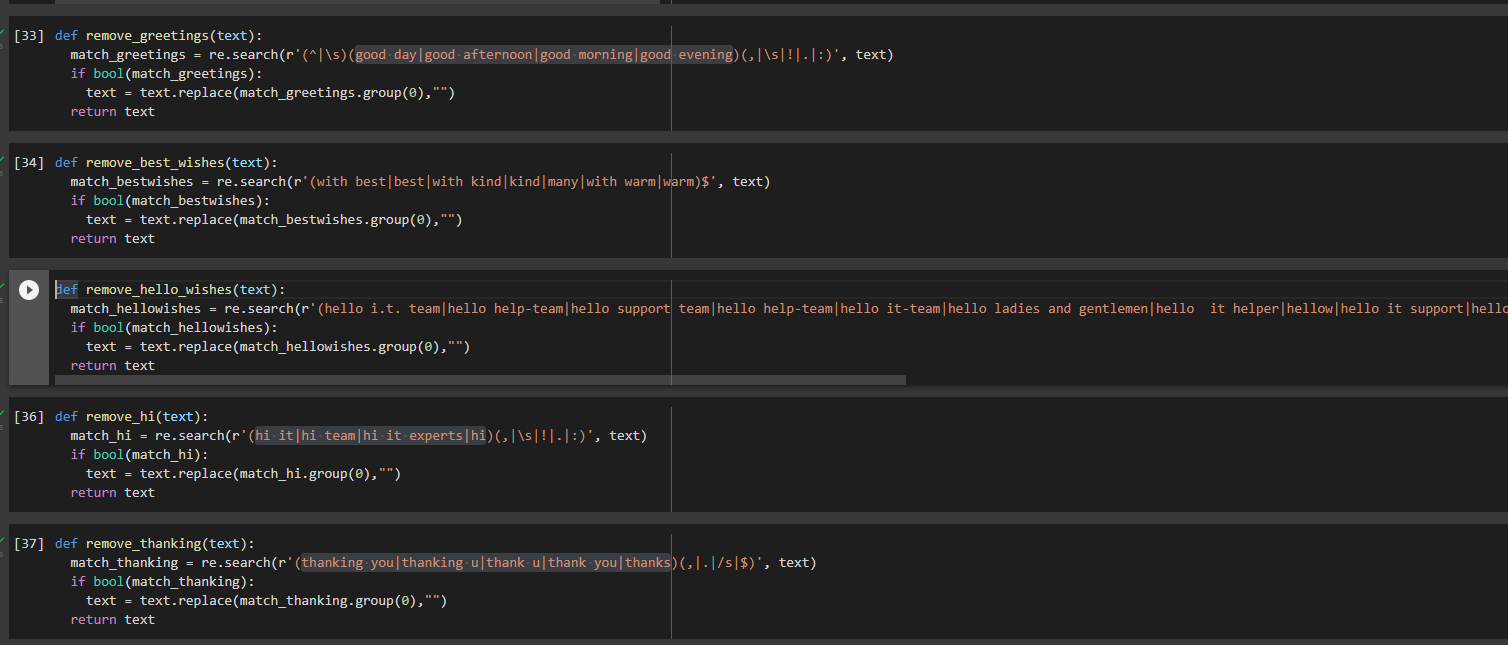


## Checking for Patterns: Phrases

Certain pattern of text found in corpus

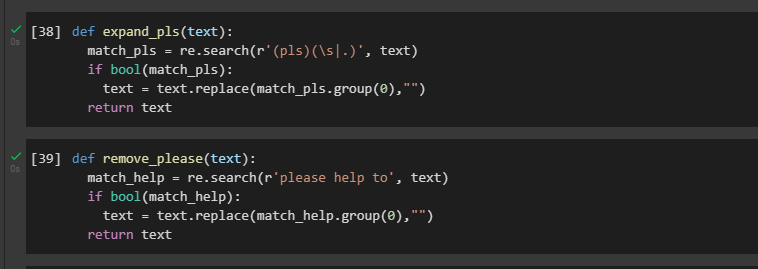
* begin forwarded message:
* sent from my iphone
* sent from my ipad
* “sir or madam,” or “sir/mam,” or “sir,”
* yes/no/na
* good day or good afternoon or good morning or good evening
* hello i.t. team or hello help-team or hello support team or hello help-team or hello it-team or hello ladies and gentlemen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



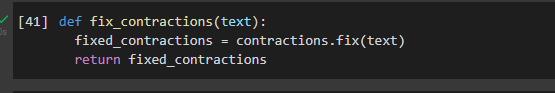
## Checking for Patterns: Expanding acronyms such as “pls” and replacing phrase such as “please help to”

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

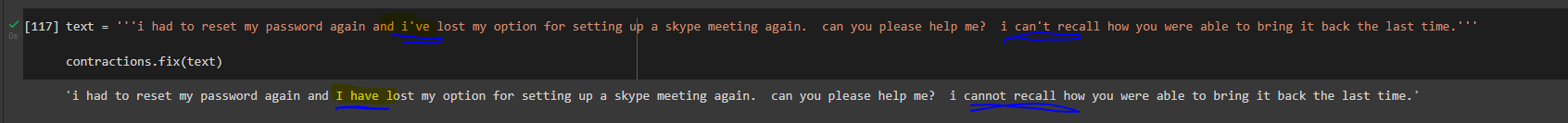


## Expanding contractions

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**

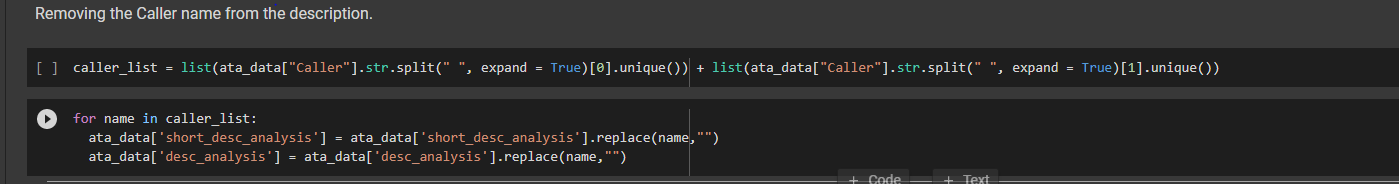


Applying contractions.fix



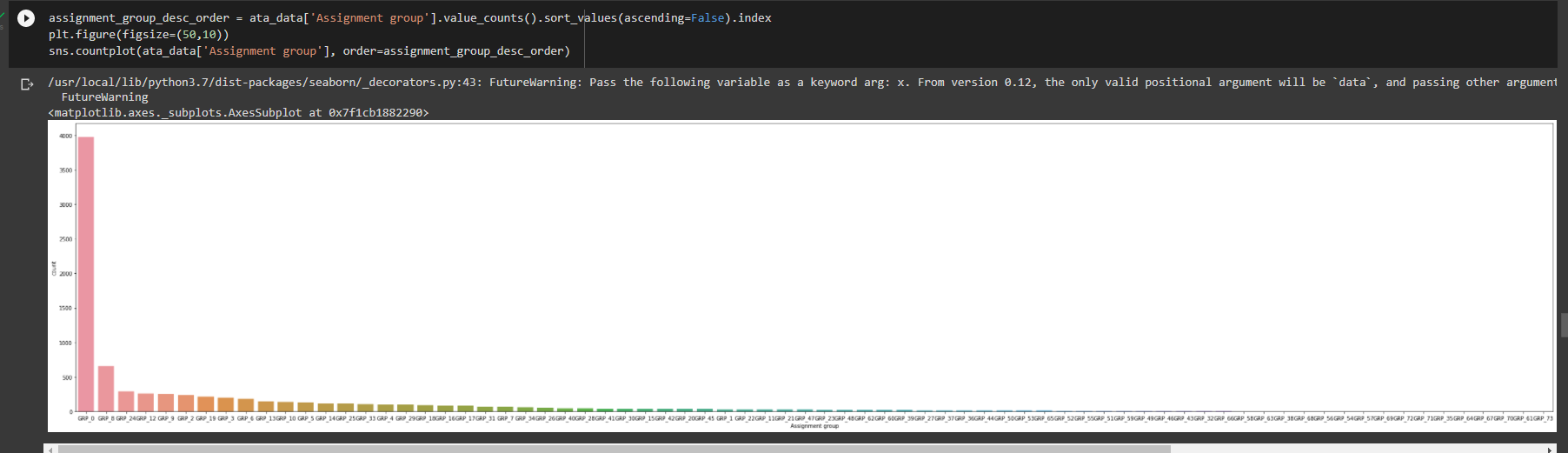
## Removing Caller’s name

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



## **Analysis of the data**

## Distribution of tickets as per Assignment Group

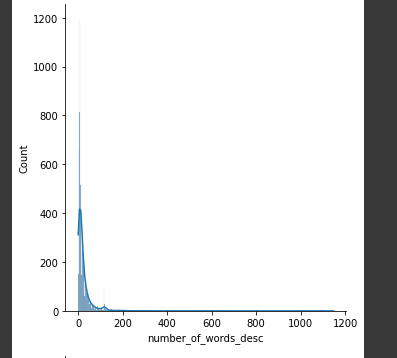


Group\_0 has 3976 tickets (roughly 47% of tickets). Dataset is imbalanced

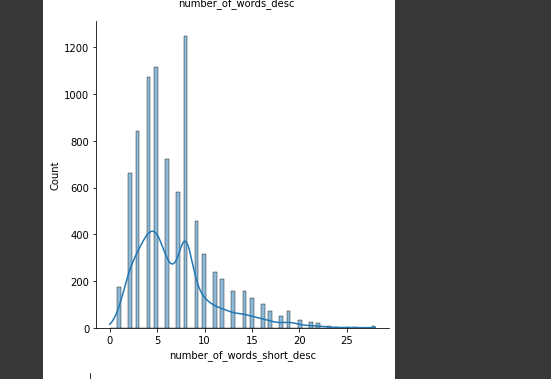
* no. of group less than 10 tickets- 25
* no. of groups in which records between 10 and 100 - 37
* no. of group greater than 100 tickets – 11

## Words/Characters analysis

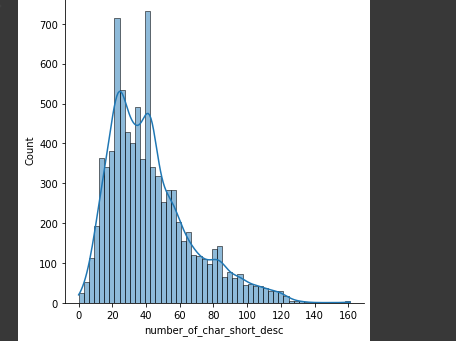
Majority of tickets have upto 100 words in Description. However, there are some tickets ranging from 200 – 1200 words



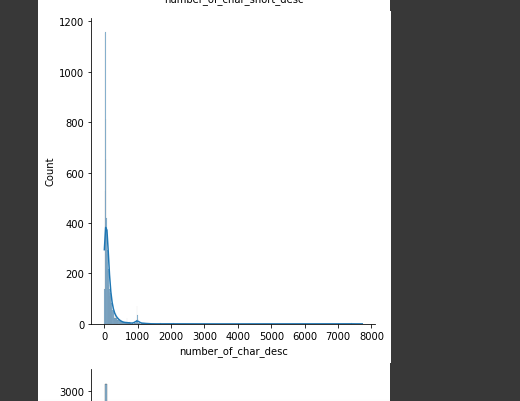
Short Description has a much compact word distribution



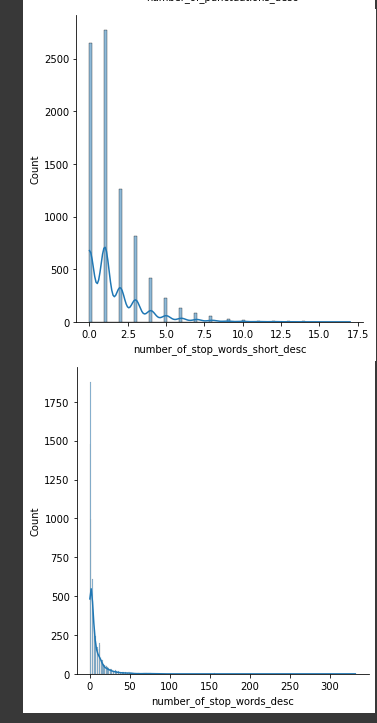
Character distribution for Short Description is also very compact



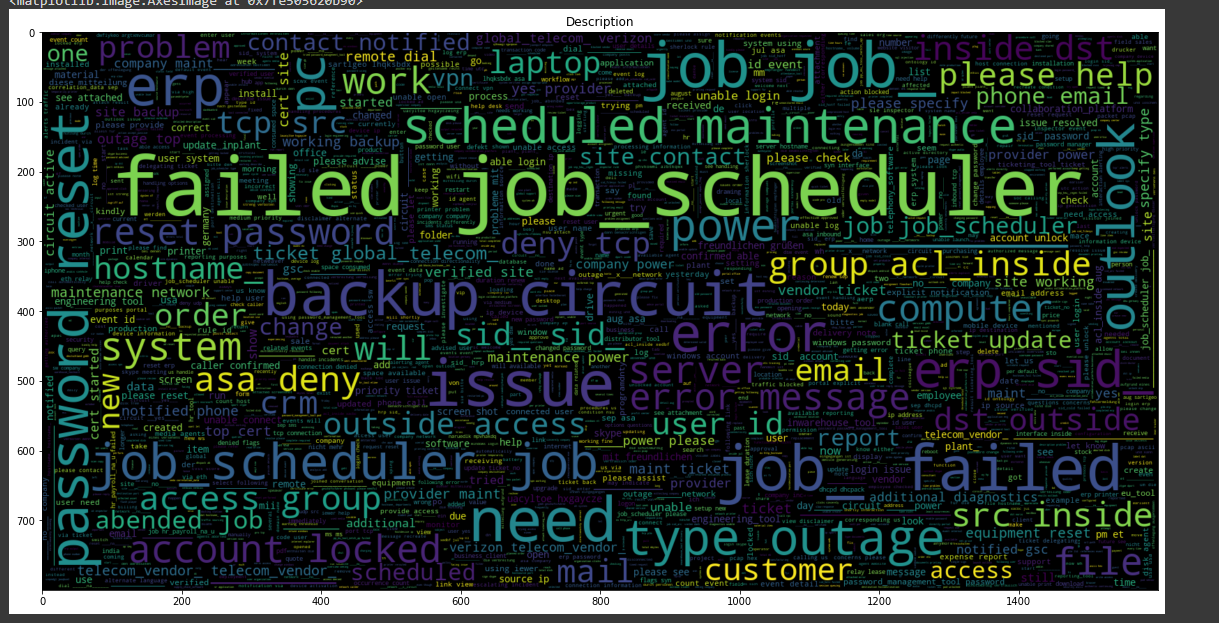
For description, in case of some tickets the character length ranges from 1000-8000 characters



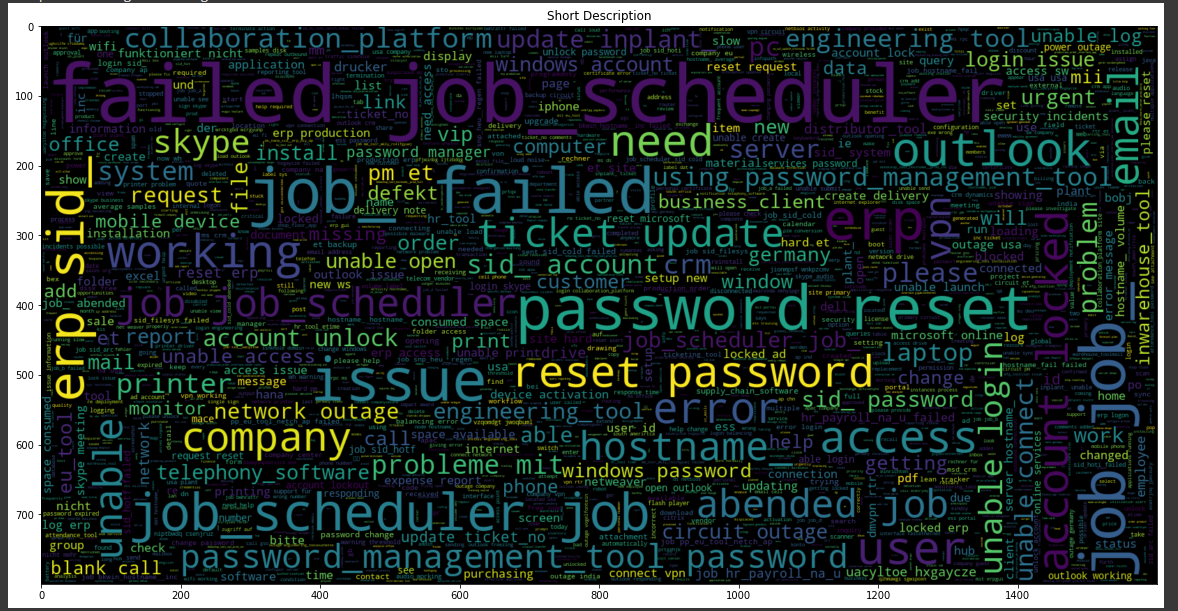
The presence of stop words is also significant in Description



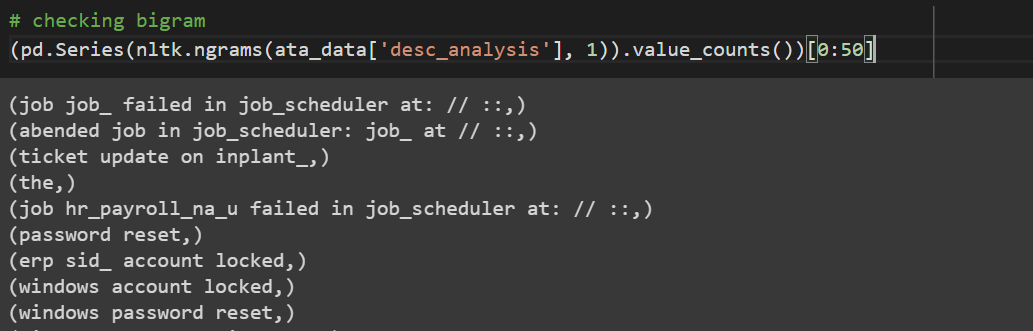
## Word Cloud of Description (whole data set) for max 5000 words



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



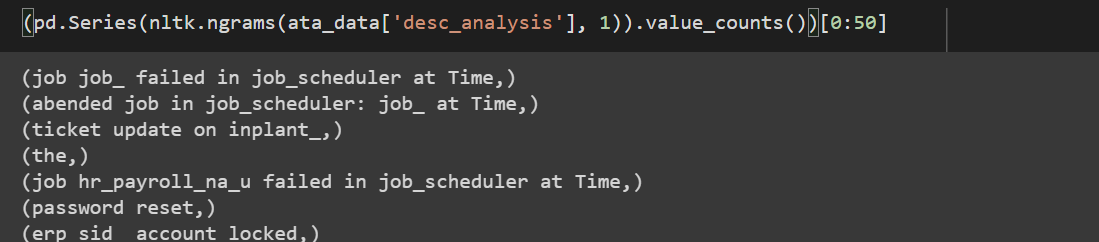
## Analyzing Ngram:



# 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



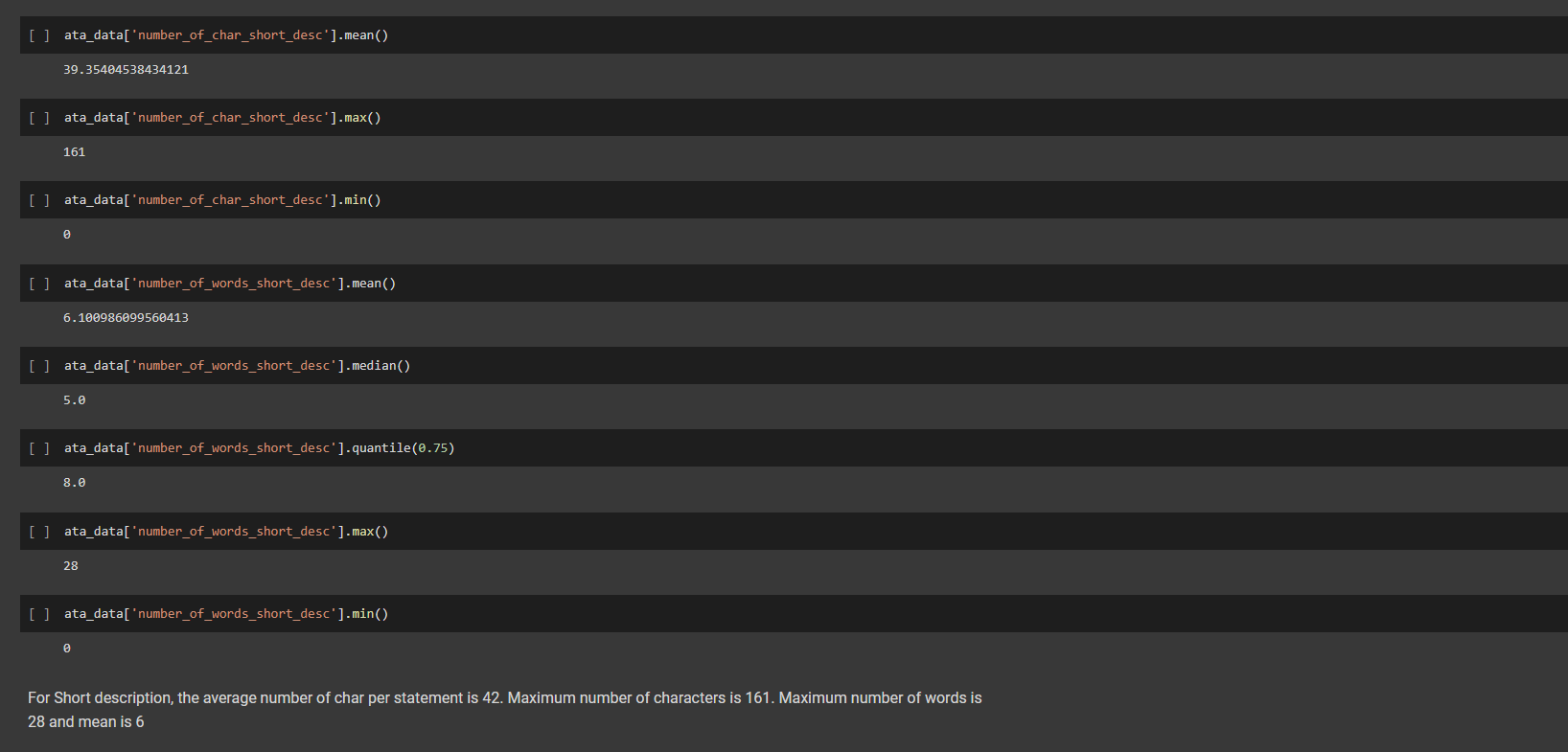
For GRP\_64, below are most correlated unigram and bigram, these are similar to GRP\_0. These two group can be merged together.



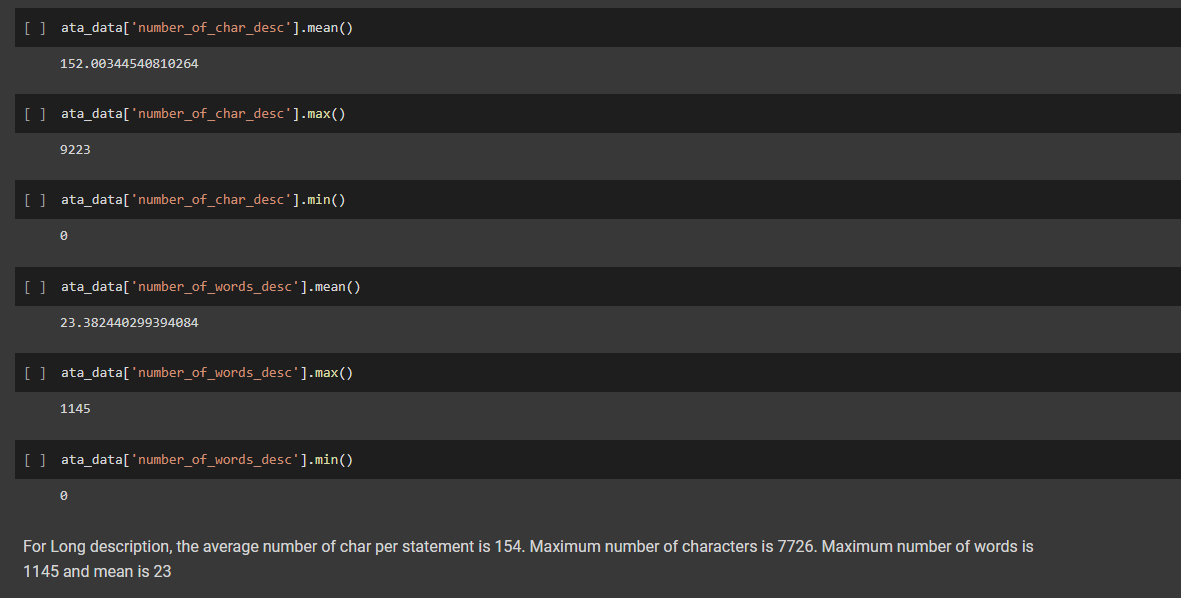
After applying above analysis (on both description and short description) on whole dataset we found these assignment groups can be merged together.

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

## 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

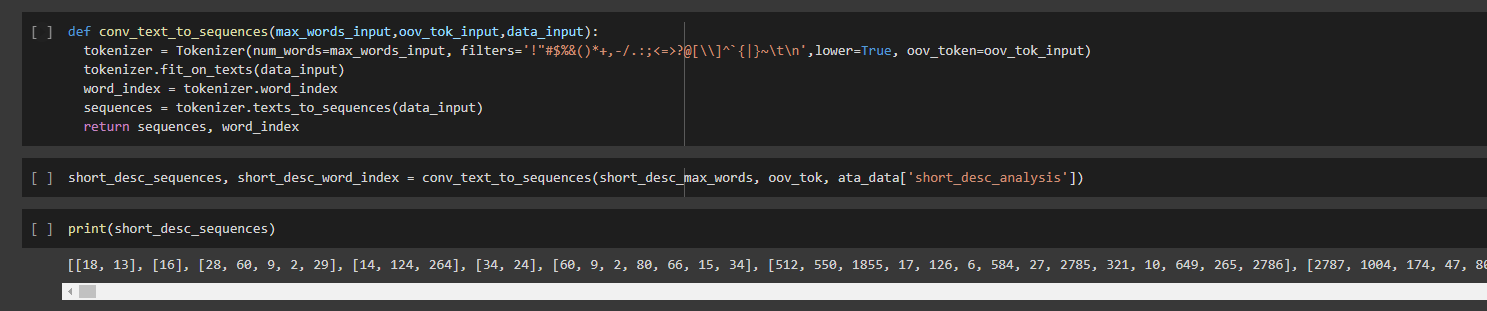
UI is created using flask.

# Model evaluation

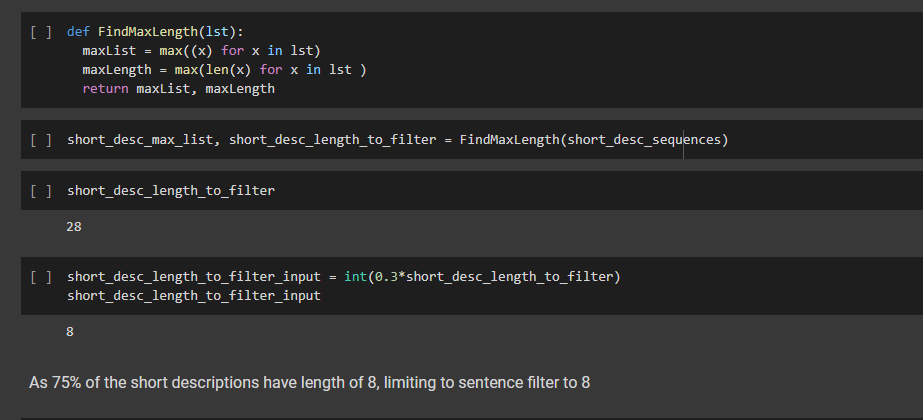
The final model selected is Bi-Directional LSTM with CNN. Glove embedding of 50D was used to create the embedding matrix.

## Process flow for the model development

## Tokenization and Text to Sequence

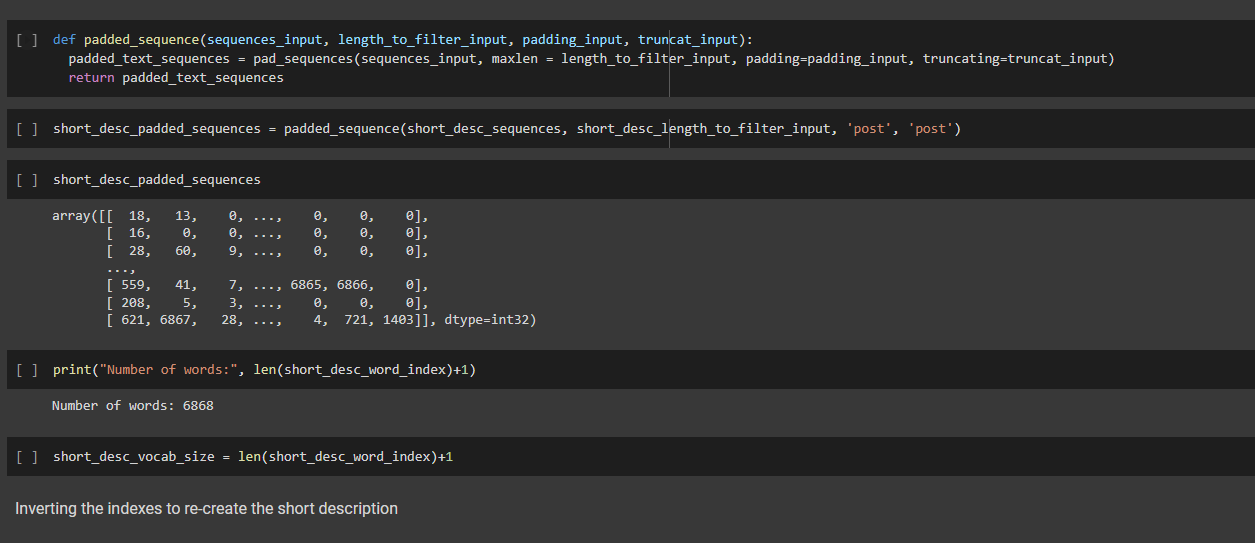


## Getting max length of a sequence



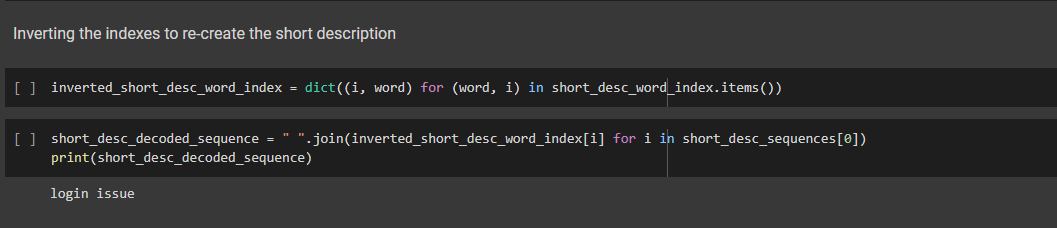
## Padding the sequence

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.



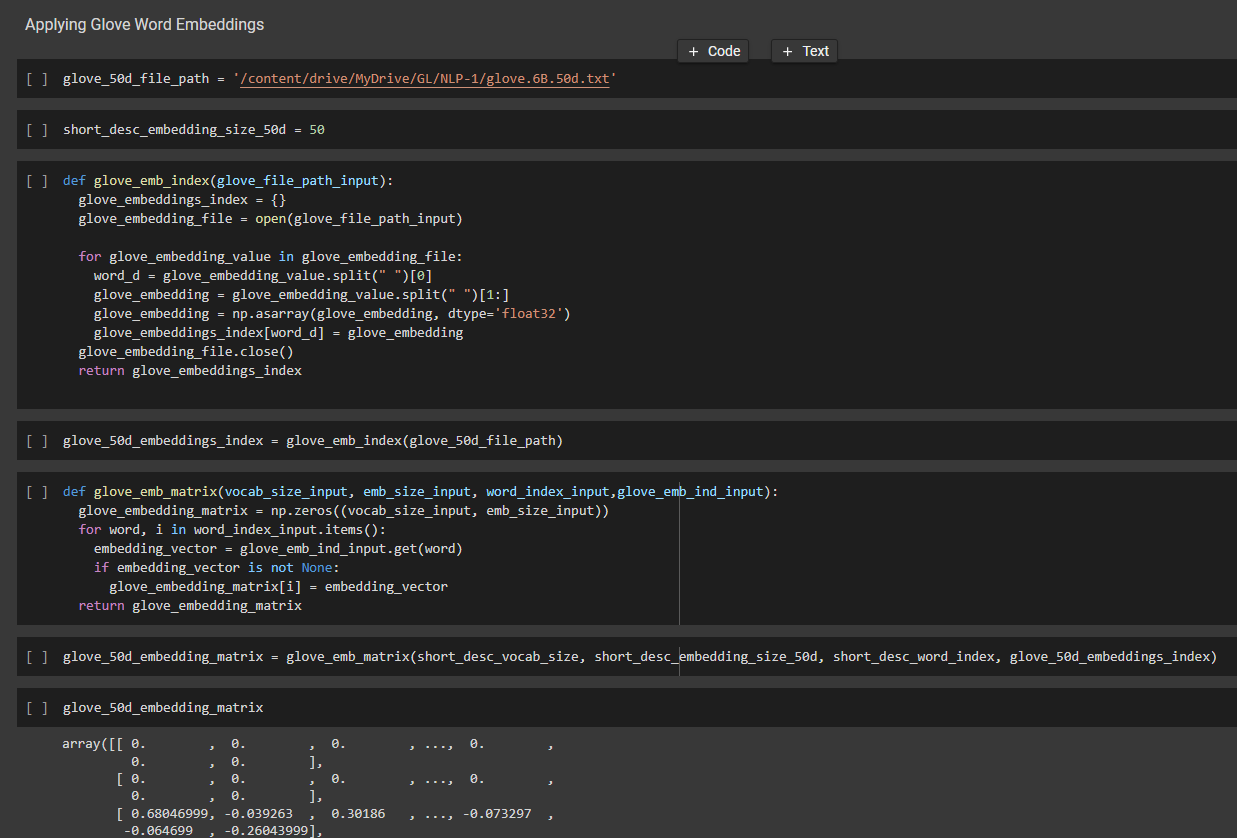
Vocabulary size is also determined.

## Inverting the word index to recreate a short description

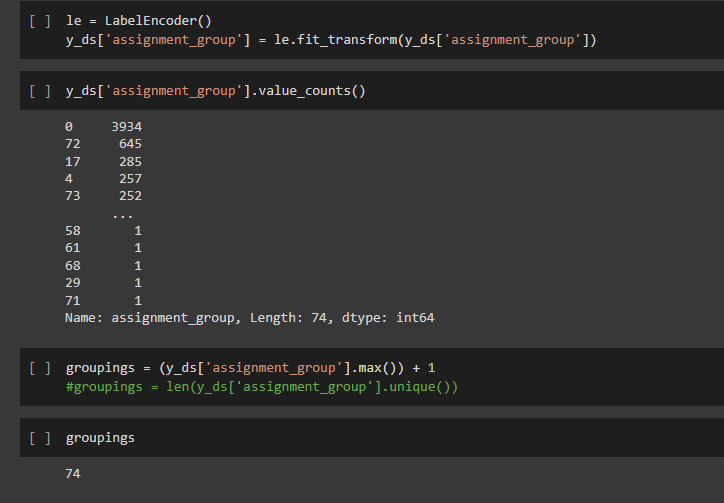


## Applying Glove embeddings

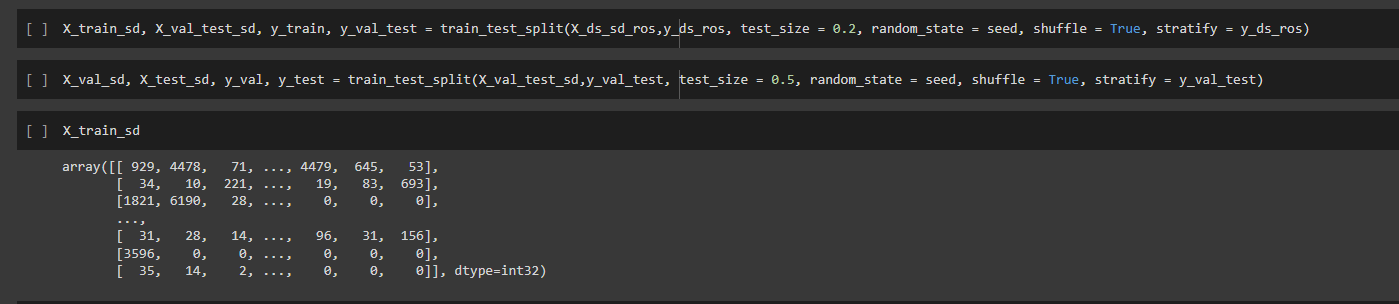
50D glove embeddings are used.



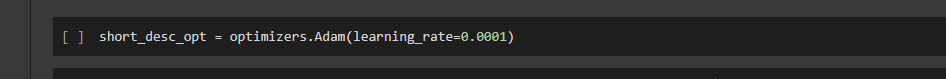
## Label Encoding to encode the target



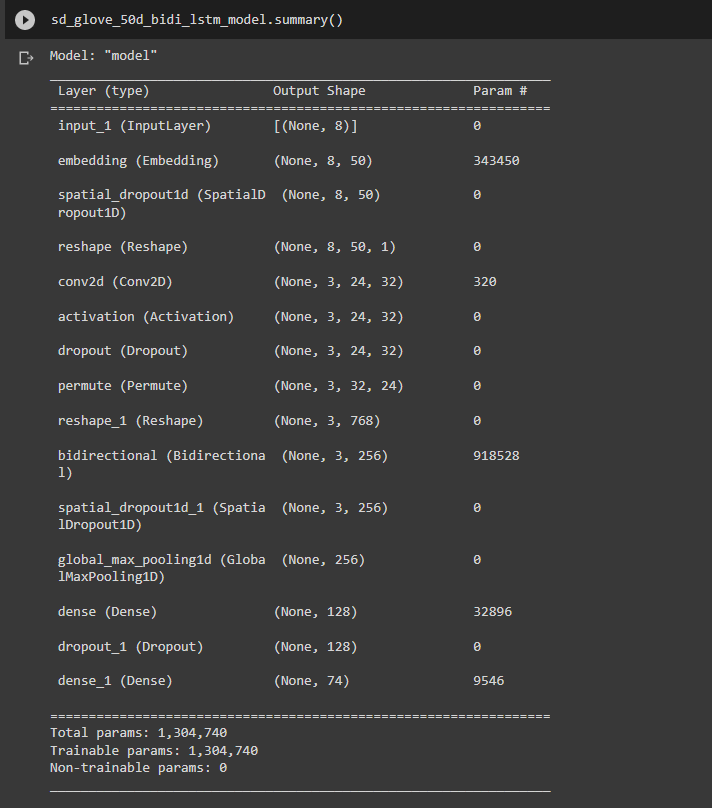
## Train Val Test split



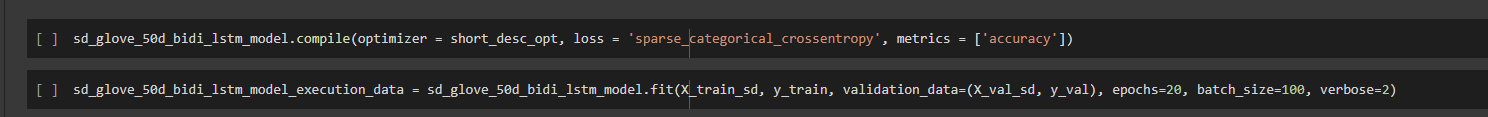
## Optimizer



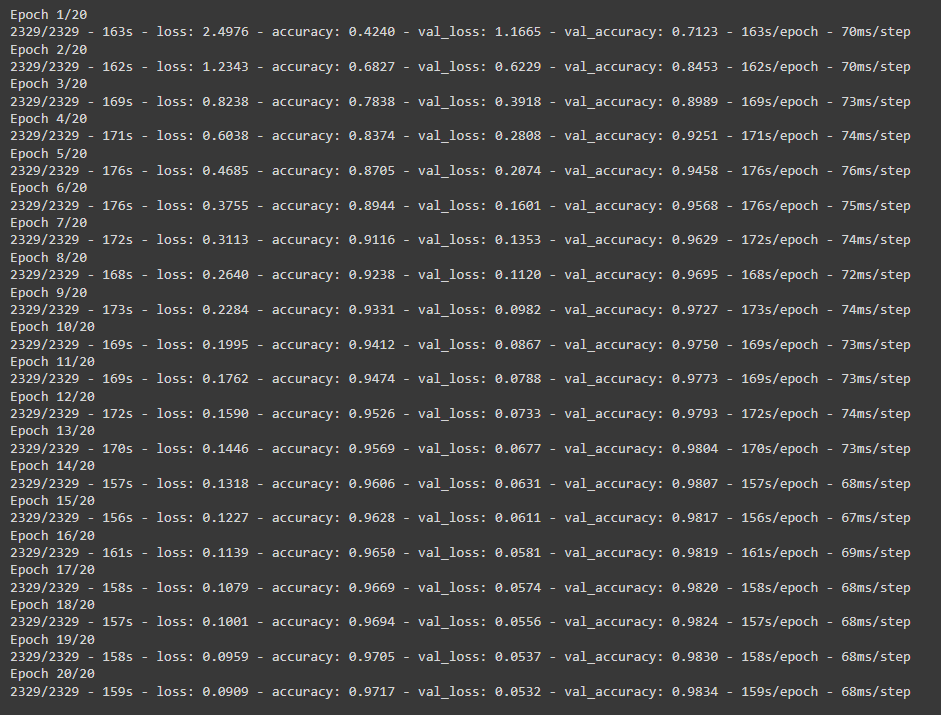
## Model Summary



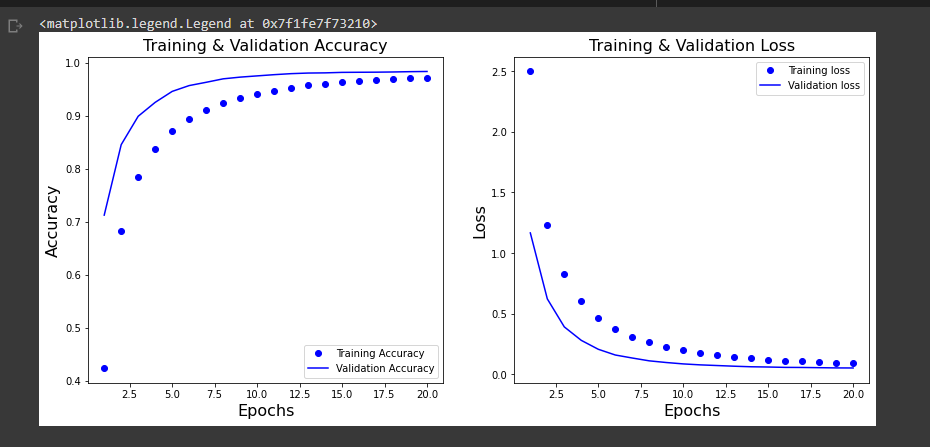
## Model Compilation and Execution



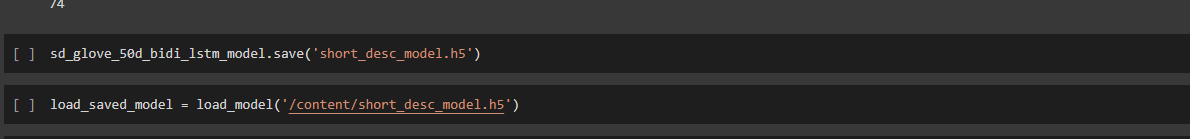
## Execution Results



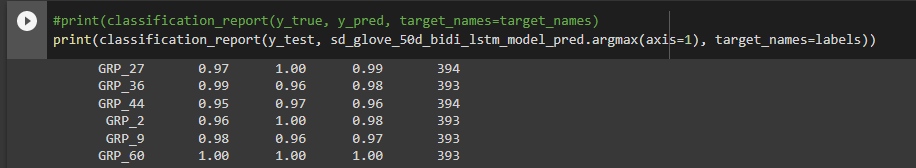
## Training & Validation Loss and Accuracy plots

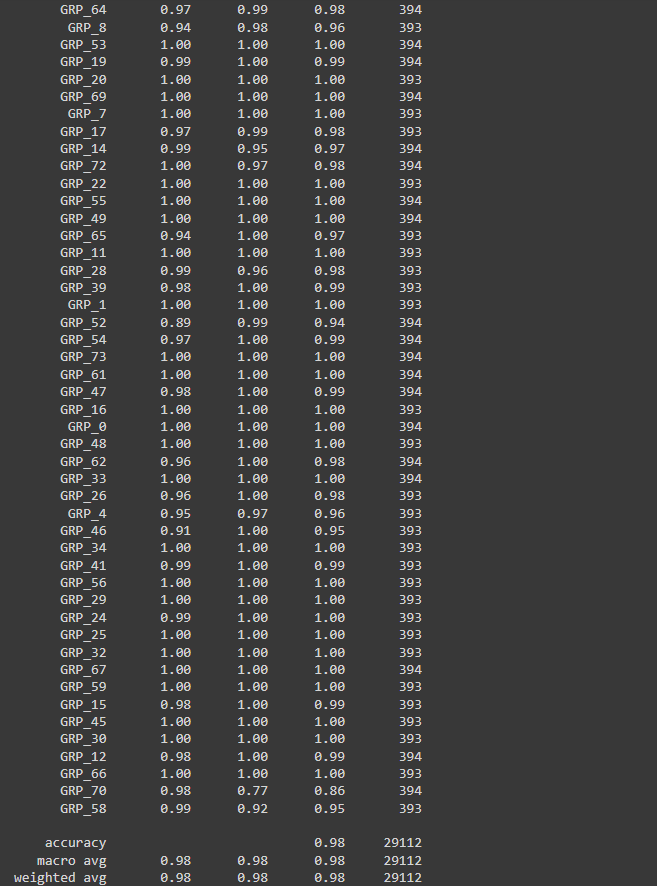


## Saving Model so that it can be used in UI

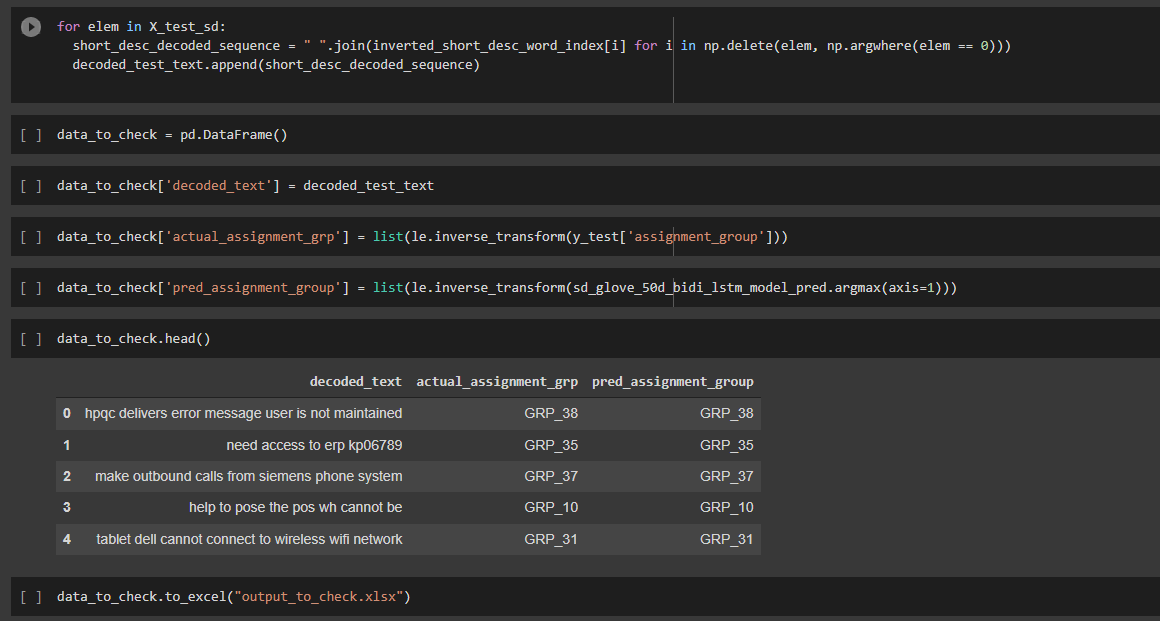


## Classification Report





Checking the final classifications

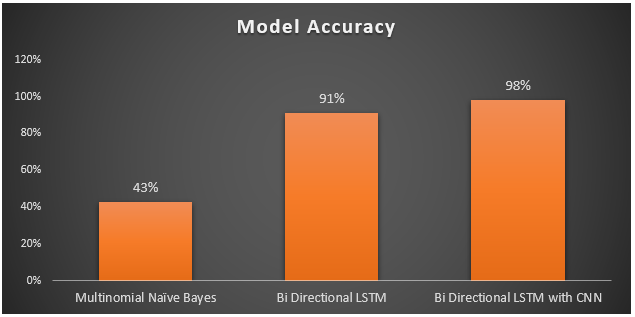


## Summary

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

# 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.

# 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.



# 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.