Automatic Ticket Assignment

# ABSTRACT

In today's Information Technology driven world, data coupled with more focus on Support processes has created opportunities for business innovation with the help of Cognitive computing techniques that aid to implement robust and enterprise centric automation. These automation's were intended to improve productivity across multiple functions and the key to such automations is human brain simulation that use the experience of the past and use that knowledge to make structured and informed business decisions.

Natural language processing is one of the main abilities of a cognitive computing system which can process the language we use and convert the text to structured data. Such an attempt is not easy to reach and involve lot of data analyzing\processing which is a complex and resource-intensive task. A huge amount of information is hidden within textual data repositories under the guise of emails, documents, chats, etc., an attempt to unwrap the hidden information can help organizations to develop better and reliable business strategies. While these techniques have wide spread applications in various business domains, problem solving besides reducing the administrative tasks are a key application.

In an Information Technology setup, IT service management (ITSM) plays an important role and is characterized by adopting a process approach towards management, focusing on customer needs and IT services for customers rather than IT systems. Incident Management, a focus area under ITSM deal with the incident tickets created by various stakes of the organization with a primary goal to resolve an issue\bring back normalcy as quickly as possible. Whenever an incident is created, it reaches the Service desk team where the issue would be analyzed with the available preliminary information and then it gets routed to the respective teams\group of the business function to resolve the issue. While the preliminary manual analysis of issue details is resource intensive and prone to human error in which case may cause further delay in issue resolution and defeat the purpose of ITSM.

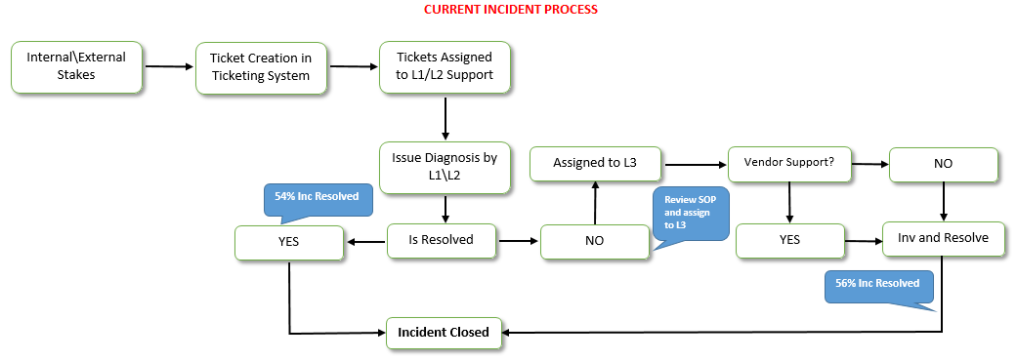
Advancements in Cognitive computing systems came to the rescue of business organizations where processes are being reviewed, automated where possible to reduce the effective cost to organization which is a key business driver in current situation.

Current study is focused in identifying the business problem in Incident Management\Support Ticket Management and based on the business drivers, built a solution which would automate ticket assignment by leveraging the NLP tech enablers.

# INTRODUCTION

In an support setup, incidents raised by various internal and external stakes were analyzed by L1 and L2 Support teams based on the incident description or available details shared by the ticket creator and resolution provided or escalated further by means of ticket assignment to teams specialized in dealing with specific problems for quick resolution of the incidents. At the outset, not all incidents created were alike due to the fact that the extent of information a specific incident may carry may be limited or not easily understood to be processed although the resolution may be the same. The first step to quick incident resolution is right ticket assignment to the team of engineers who are equipped or proficient in dealing a specific category of issues that fall under their scope. While the correct ticket assignment can help reduce the operational cost noticeably and also help use the valuable resources on more productive work to improve the overall service experience.

Below is the Current Incident Management Process –



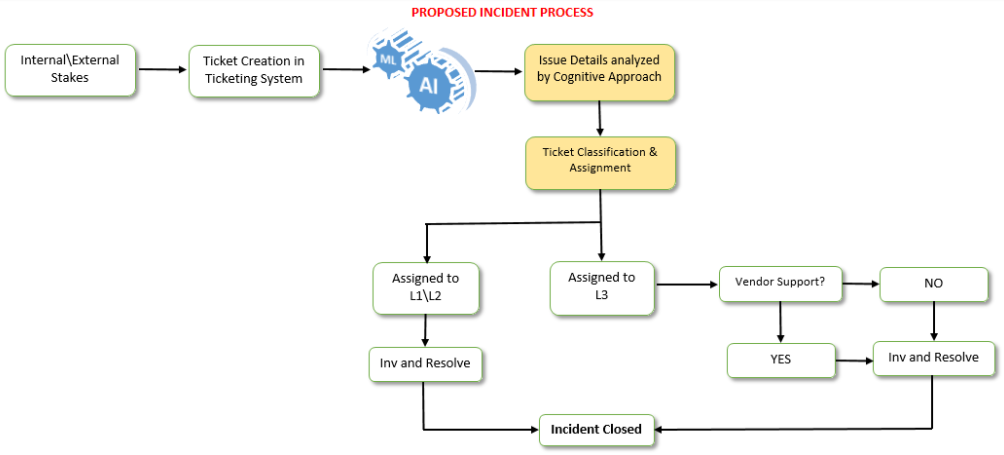
# BUSINESS PROBLEM

Above current process reveal that incidents are created by various stakeholders 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 L3 teams after reviewing Standard Operating Procedures (SOPs) which is time consuming and ~25-30% of incidents needs to be reviewed for SOPs before ticket assignment. At the same time, not every instance of ticket assignment by L1/L2 teams are accurate and ~25% of incidents are wrongly assigned to functional teams. L3 teams will carry out detailed diagnosis and resolves the incidents. Around ~56% of incidents are resolved by Functional / L3 teams. In case, vendor support is needed, they will reach out for their support towards incident closure. As the ticket assignment process is manual, instances of unattended or timely action on incidents raised by stake holders degrade the effectiveness of the support eco system.

# SOLUTION

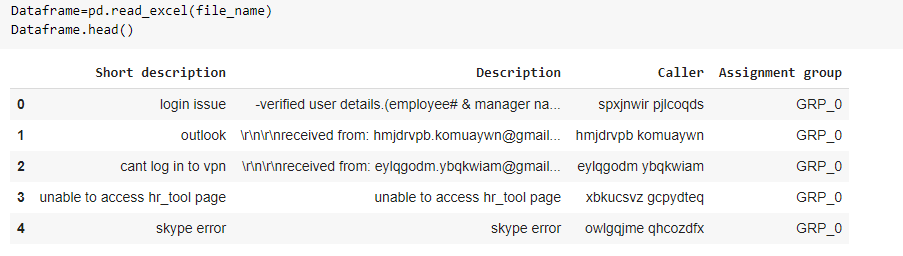
The solution to the Business Problem is the Proposed Incident Management process where the current incident creation step shall remain as-is and are fed to the Cognitive techniques like NLP aided by Machine Learning Algorithms where the details of the incidents were analyzed by the models and classification of the incidents were performed based on prior experience. The output from the Machine Learning model will be the predicted Classification group or the Support Group responsible for solving similar incidents in the past. Based on the output of the models, the incident shall be assigned to respective Assignment Group as classified by the Machine Learning model. Post ticket assignment, the issue resolution step can remain as-is.

Proposed Incident Management Process –

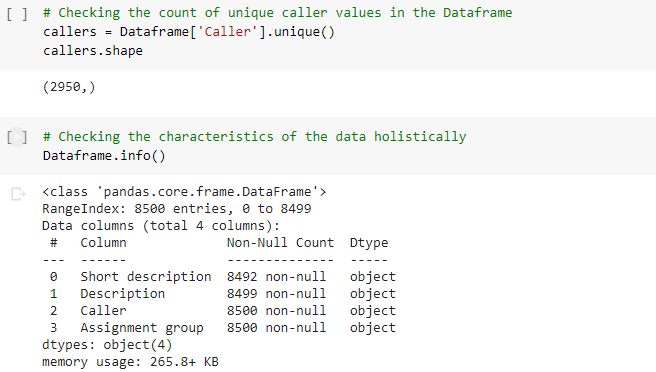


# EDA and Pre Processing

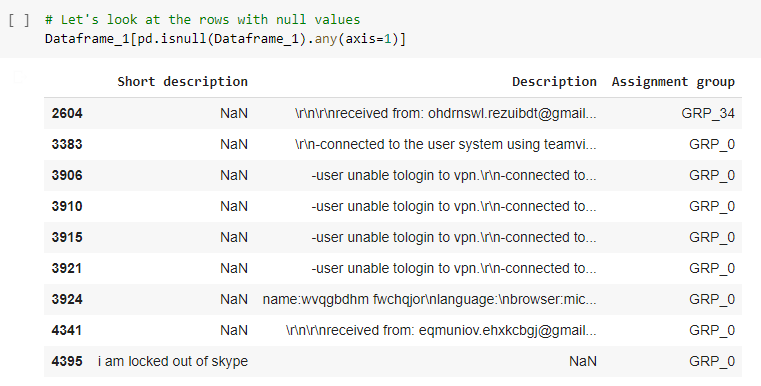
To build the model and validate them we need to understand the given dataset. On analyzing we see the given dataset has the attributes four different attributes and total of 8500 records all are Object data type.

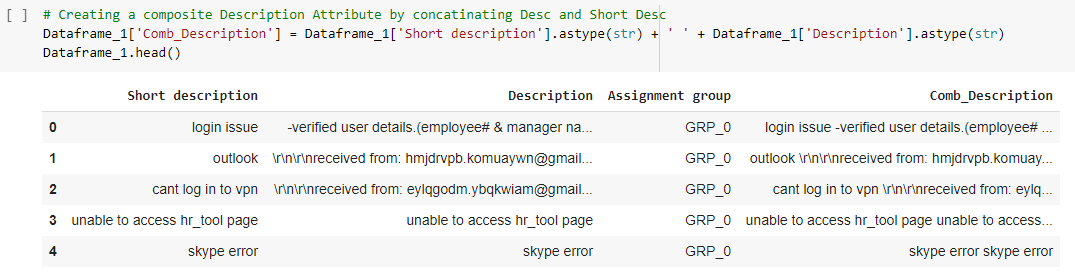


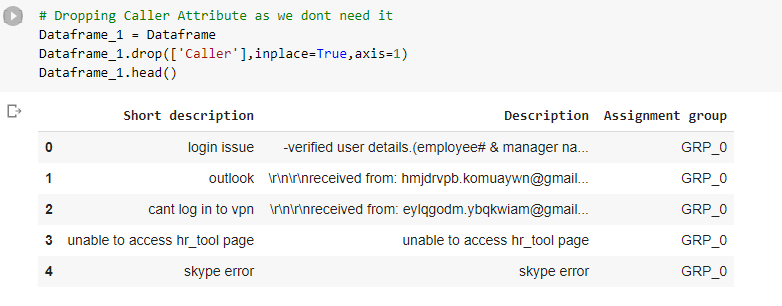
As we see from head (top 5 records) of file above the four attributes are as Short Description, Description, Caller, Assignment group. Out of these one is the target attribute (Assignment group). To check on the count of the records in the given dataset and datatypes of each attribute we did below steps:



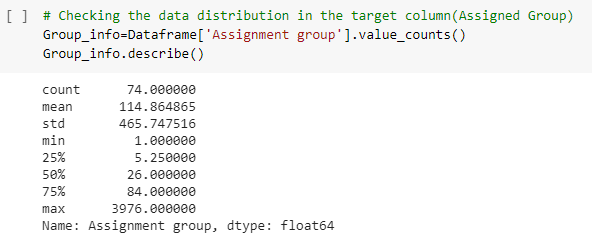
From above results we can infer that for few records either description or short description is blank. So we decided to combine the two attributes so that we come over the blank value problem. Also from further analysis of data we took decision to drop the attribute: Caller. It doesn’t contribute much to the classification logic.

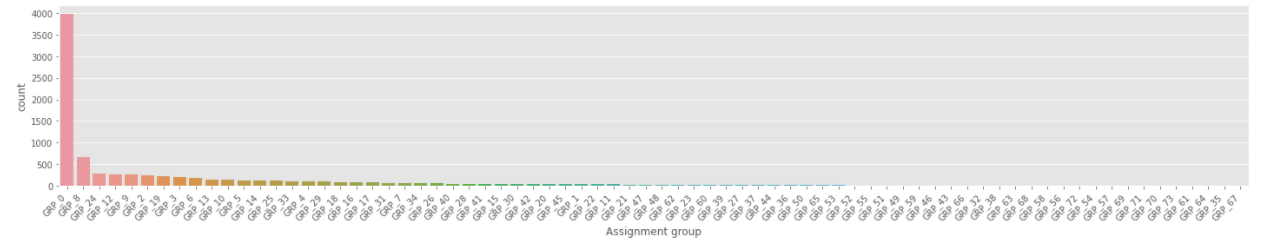




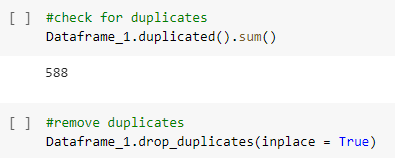


Data distribution analysis:

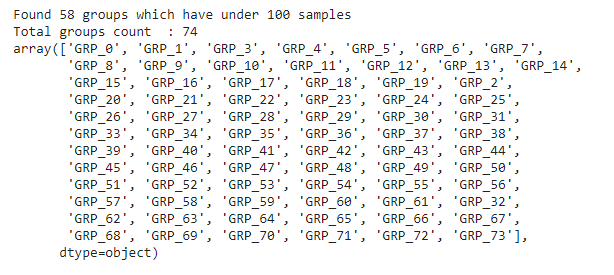




Also checked on the duplicate data which were figured out as 588 records. So re moved the duplicate data.

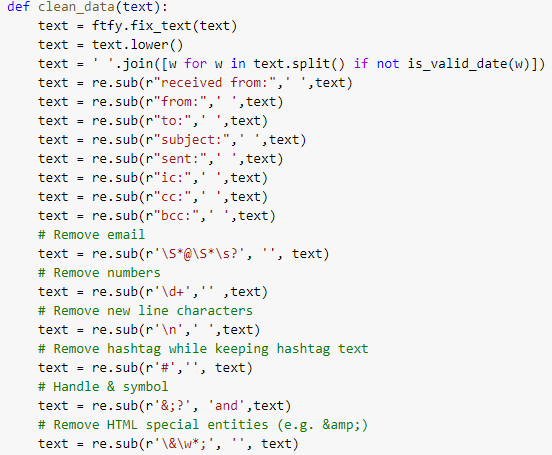


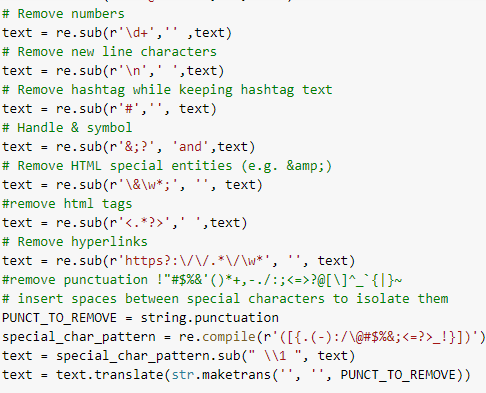
We see that the data has distinct 74 groups in total. Data is extremely skewed as one of the group counts to about 4k records where as mean of complete data set is 114 with standard deviation of 465. Further analysis of number of groups which have the record counts less than 100 so that we can seek the opportunity to merge the similar groups. We found that out of 74 groups we have 58 groups which have less than 100 samples.



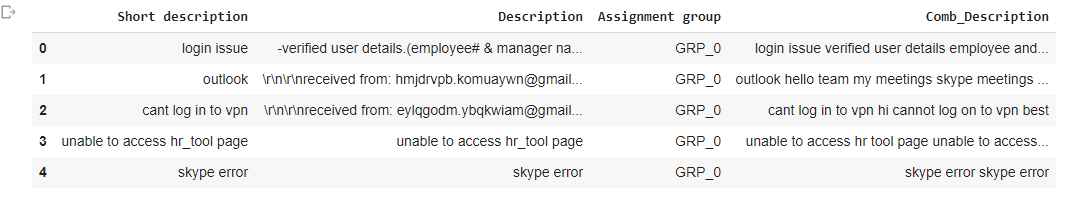
Data Cleaning

After these preliminary steps when we take a look at data we can still see a need of preprocessing which will include the removal of stop word and few sequences as identified the mail senders and the punctuation marks. The complete list of removals is in the code which is submitted along with this doc. (Few screenshots of the same are as: )



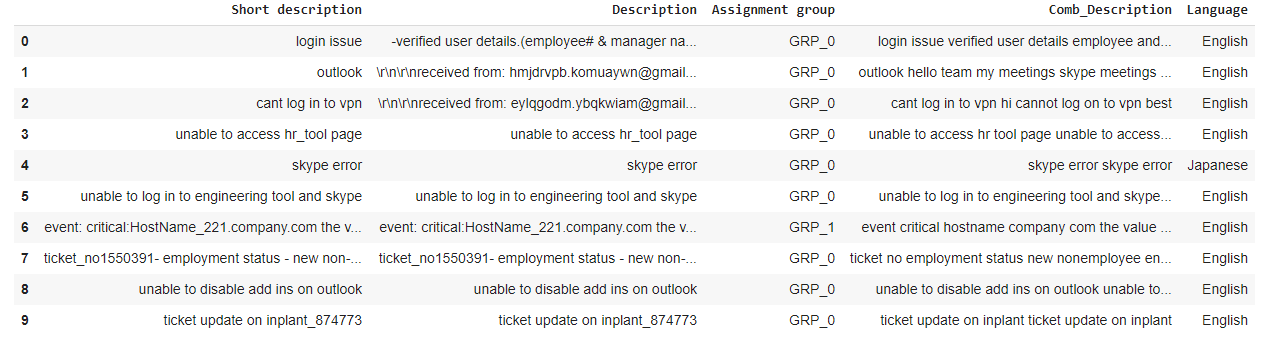


Post this processing of clean data function we see a better formatted data as:

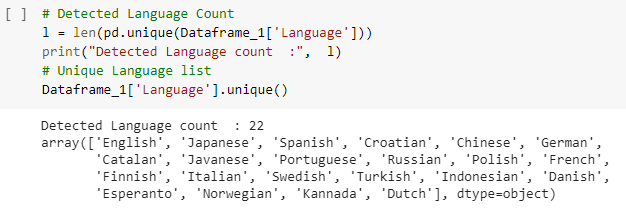


Language Detection

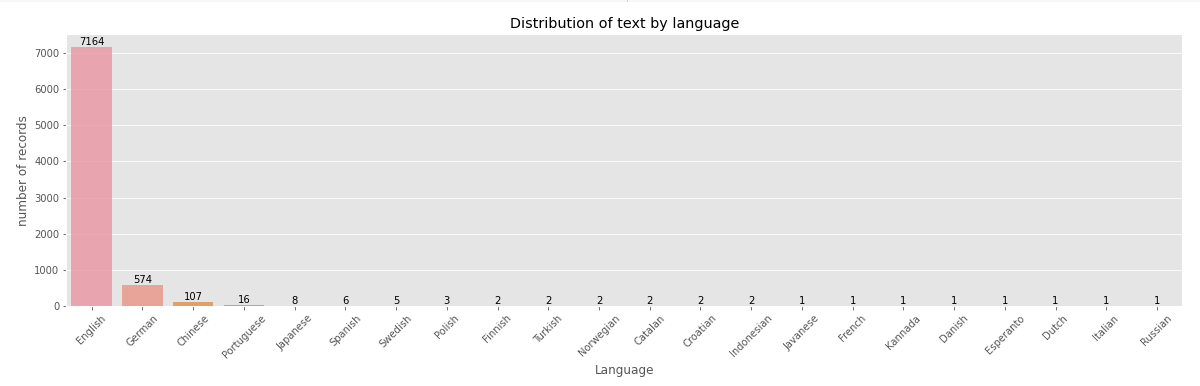
Now the data seems ready to be identified in which all languages is it in. We used a pre trained model to detect the languages of the combined description and convert them to English as a common language. The result of it was as below:



Total 22 languages were detected in the complete dataset:

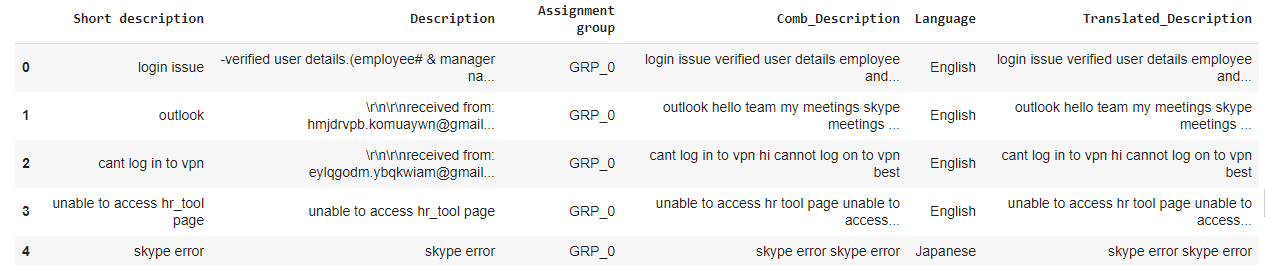


Data distribution among the different languages is as:

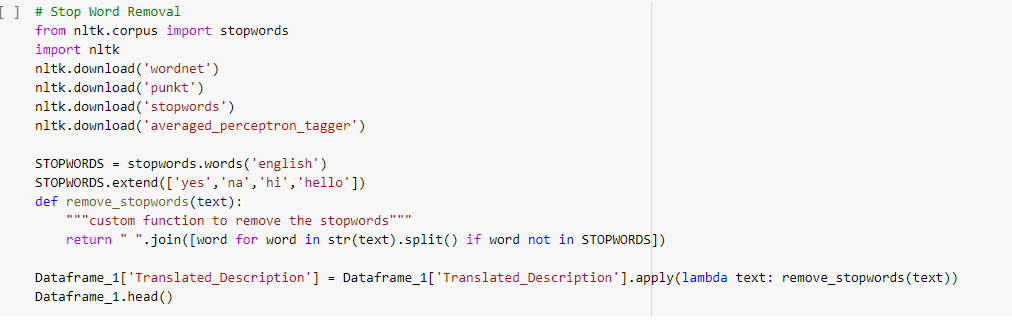


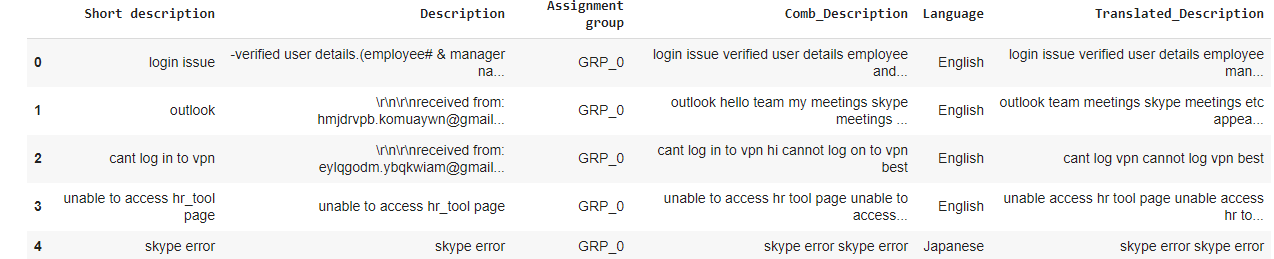
Language Translation

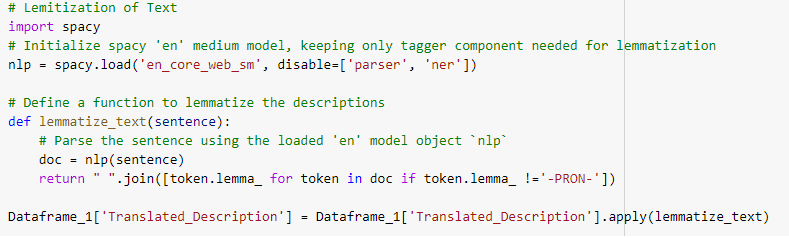
We see evidently most records are in English. Hence we translated all other languages to English using google translator. The translated data set looks like:



We again performed the preprocessing of the translated data to remove the stop word by extending to a few more words. This is followed by the lemmatization:







Getting Word Cloud

Post these preprocessing we created the word cloud of few major groups to get an idea of major key words in each group.

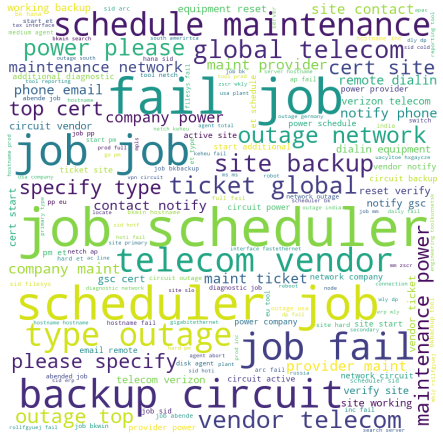
This is the word cloud of whole dataset



This is word cloud of Group0:



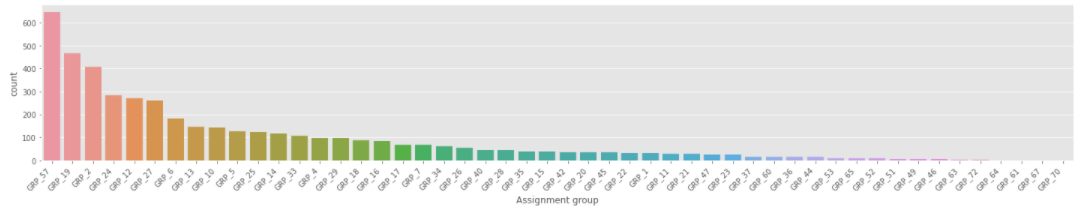
This word cloud is of the group8:



From above three word clouds we see some of the evident words in the whole groups like Scheduler, Job, telecom, fail, file, access, and password.

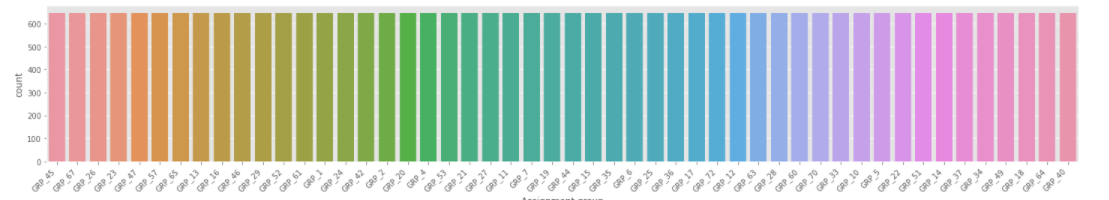
Identifying the similar groups

Since there are few groups where we have very less records so we decided to find some similarity across groups using few techniques (like fuzzy wuzzy / correlation). Based on the similarity percentages we can merge the groups so that we don’t loose on data and need not up-sample a lot of groups as well. On using the similarity percentages among group and merging them with a threshold of 80% we finally get total 51 groups. The distribution of data among them (except group0) is as below:

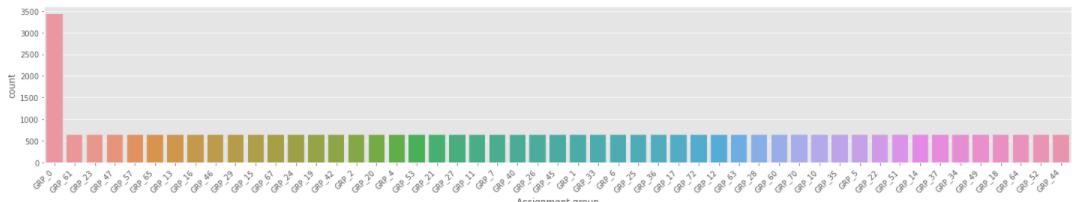


Data Sampling

Now among these merged 50 groups we applied the decided to apply the up-sampling of data with a defined random state. The visualization of the up-sampled data except group0 is as:



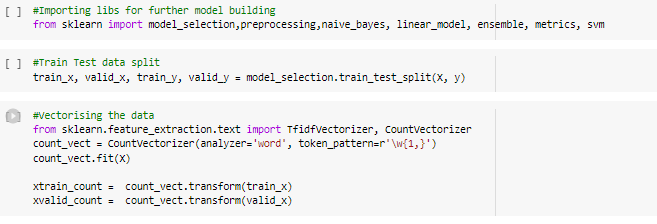
Visualization of the complete data set is as:



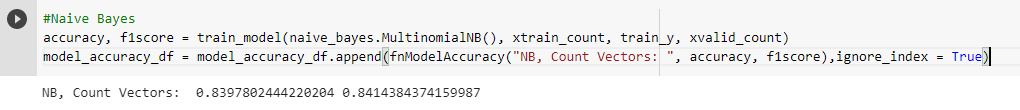
# Model Building

Now with preprocessed data we are ready to build few models on it and test their accuracy. Since this is a classification problem we decided to implement four models. Compared their accuracy and F1 scores on test and train split data.

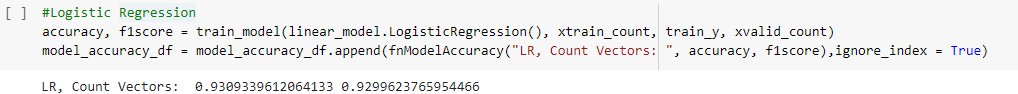
Firstly we split the data set in Test and Train and vectorized it:



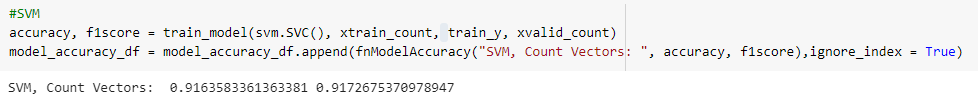
**Model1: Naive Bayes Model**



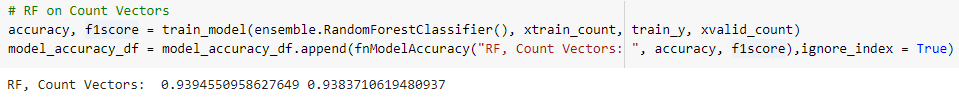
**Model2: Logistic Regression**



**Model3: SVM**

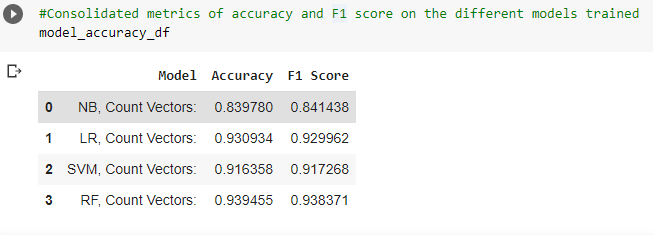


**Model4: Random Forest**



# Model Performance

By implementing these four models we get the final scores for comparison of their performances. The consolidated metrics to compare their performances is as:



From the above matrix we can deduce that Random Forest and Logistic Regression models are much better performer in our case. But the noticeable point here is that we have not yet tweaked the hyper parameters of these models at all as of now.

The way ahead is to tune these models on hyper parameters and see if the performances can be improved. Also we need to seek if any neural network model can also be implemented and test its performance. Once we bring in the Neural network model we will have to also take into account the time elapsed on each model so that it is understandable where the trade-offs can be handled and which model then suits to our problem statement as best.

***NOTE:*** We have the working python code till the comparison of the machine learning models implementation and is submitted along with this doc for review.