**AIML CAPSTONE PROJECT**

**INTERIM REPORT – AUTOMATIC TICKET ASSIGNMENT**

**Team: Srikanth , Shivanshu ,Hardeep ,Rama, Gnanaskanda**

**Supervised by: Sanjay**

**Date :17 May 2020**

# INTRODUCTION

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 analysed and assessed by organization’s support teams to fulfil 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. In case 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. In case 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.

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

Project Description

In this capstone project, the goal is to build a classifier that can classify the tickets by analysing text. Details about the data and dataset files are given in below link

<https://drive.google.com/file/d/1OZNJm81JXucV3HmZroMq6qCT2m7ez7IJ>

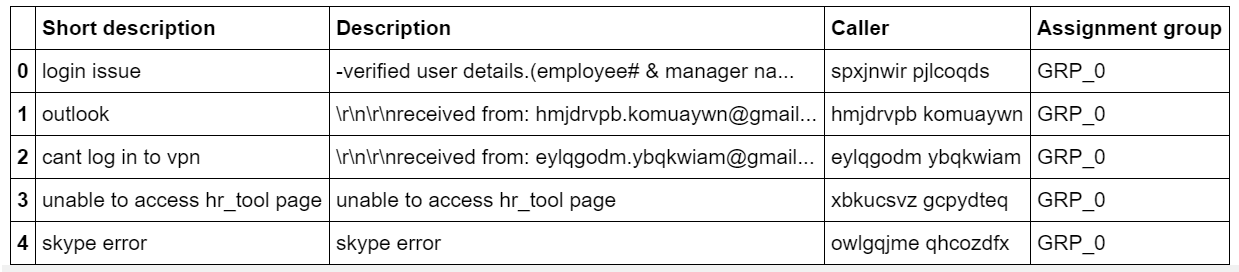
Milestone 1:

Pre-Processing, Data Visualisation and EDA Overview

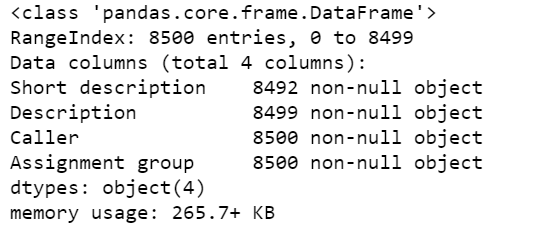
* Exploring the given Data files
* Understanding the structure of data
* Missing points in data
* Finding inconsistencies in the data
* Visualizing different patterns
* Visualizing different text features
* Dealing with missing values

## Explore the data files

It was found that the Given data consists of 8500 records with 4 Features as shown below



The concise summary of the data provided looks as below



The missing Values were found using the df.isna() of Pandas. The list of missing values is as below

- Short description 8

- Description 1

- Caller 0

- Assignment group 0

- dtype: int64

As you can see above 8 records for the short Description and 1 record for Description are missing values

Further analysing on the objects containing unique values using the df.Value\_Counts( ) it was found that this data has 74 unique Labels .

GRP\_0 3976

GRP\_8 661

GRP\_24 289

GRP\_12 257

GRP\_9 252

GRP\_2 241

GRP\_19 215

GRP\_3 200

GRP\_6 184

GRP\_13 145

GRP\_10 140

GRP\_5 129

GRP\_14 118

GRP\_25 116

GRP\_33 107

GRP\_4 100

GRP\_29 97

GRP\_18 88

GRP\_16 85

GRP\_17 81

GRP\_31 69

GRP\_7 68

GRP\_34 62

GRP\_26 56

GRP\_40 45

GRP\_28 44

GRP\_41 40

GRP\_15 39

GRP\_30 39

GRP\_42 37

...

GRP\_36 15

GRP\_44 15

GRP\_50 14

GRP\_53 11

GRP\_65 11

GRP\_52 9

GRP\_51 8

GRP\_55 8

GRP\_49 6

GRP\_59 6

GRP\_46 6

GRP\_43 5

GRP\_66 4

GRP\_32 4

GRP\_58 3

GRP\_63 3

GRP\_68 3

GRP\_56 3

GRP\_38 3

GRP\_69 2

GRP\_72 2

GRP\_71 2

GRP\_57 2

GRP\_54 2

GRP\_67 1

GRP\_64 1

GRP\_61 1

GRP\_70 1

GRP\_35 1

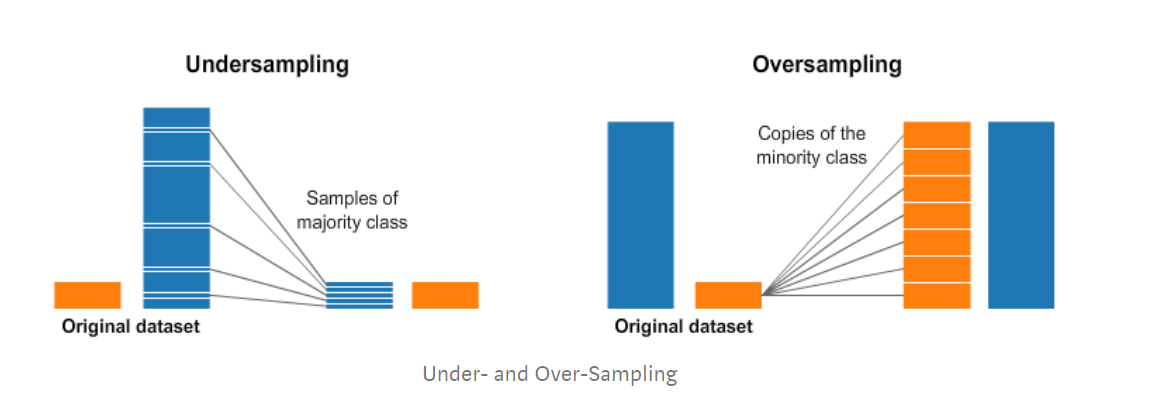
GRP\_73 1

As it is seen above this is a highly imbalanced data set and 46% of the data set is represented by a single class GRP\_0. Out of the 74 Classes 30 classes have less than 15 data points of which 19 classes have less than 5 data points

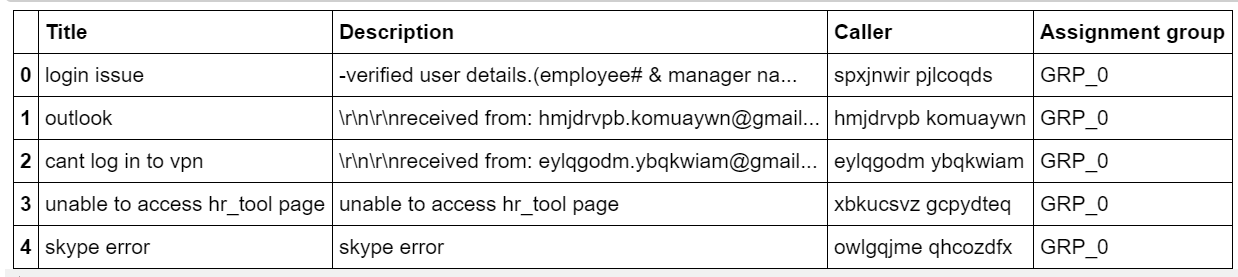
The key to building a good machine learning model is the data it is trained on. Therefore, it is imperative that the training data be clean and balanced. There are several approaches to achieve this

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. Resampling is done after the data is split into training, test and validation sets. Resampling is done only on the training set or the performance measures could get skewed. Resampling can be of two types: Over-sampling and Under-sampling. Which is the approach adopted in this case

Under sampling involves removing samples from the majority class and over-sampling involves adding more examples from the minority class . The simplest implementation of over-sampling is to duplicate random records from the minority class, which can cause overfitting. In under-sampling, the simplest technique involves removing random records from the majority class, which can cause loss of information.

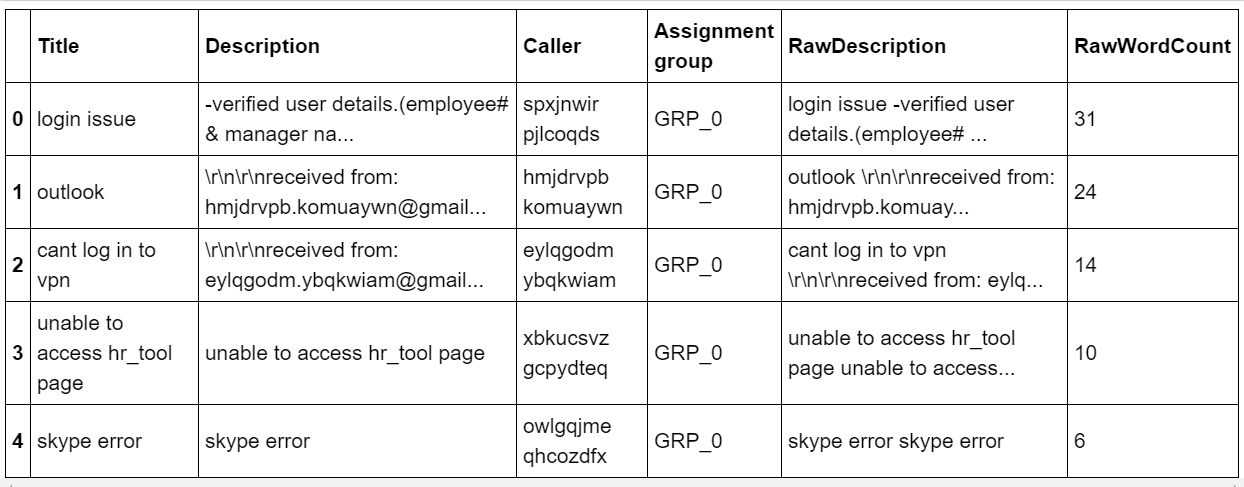


In order to further proceed and simplify the analysis the Column “Short Description” has been changed to “Title” and our data looks like below after this



* Pre-processing is an essential, wherein the text is to be cleaned up to bring it to a required format for the information extraction models. This includes normalizing different tenses of words, normalizing synonyms, spell correction etc.
* Further, it is necessary to segregate the mails and identify if a block of text is header or signature, email address, user name or the body of the mail
* Spell correction and normalization of abbreviations are also required, using regular expressions.
* This processed text then needs to be tokenized, which is to split the raw text to a list of words, using popular open source libraries like nltk and spaCy.

Let us look at the Raw word count across before pre-processing

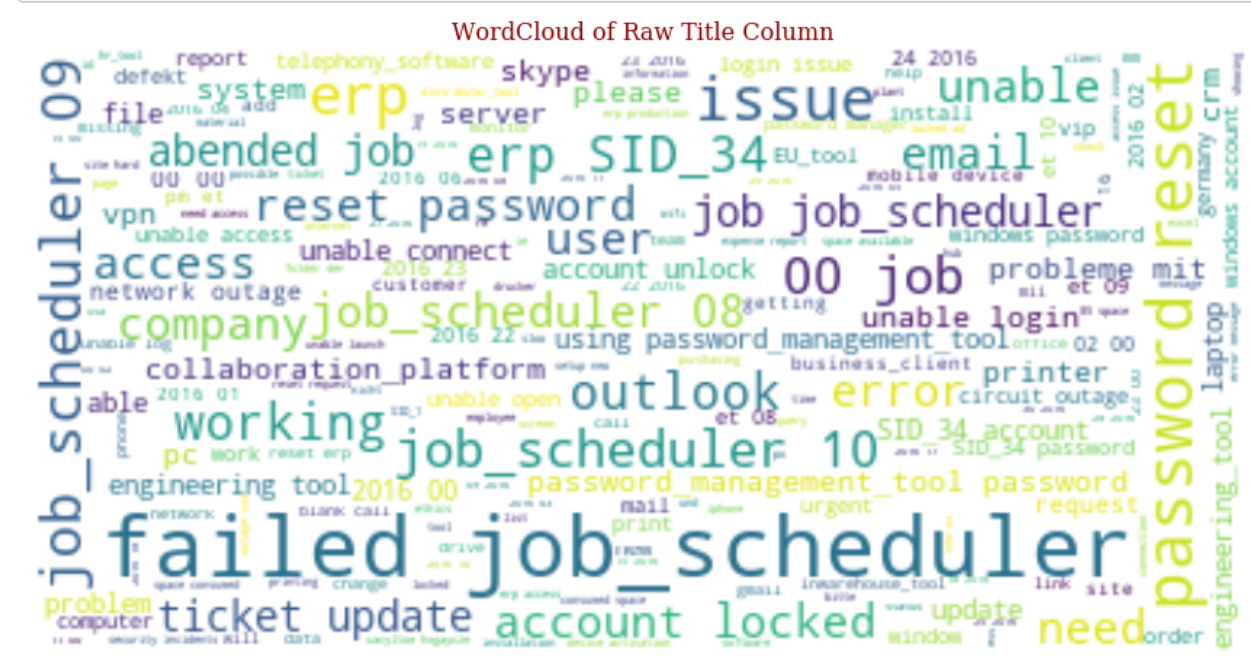


As we can see above are the Raw word counts before pre-processing

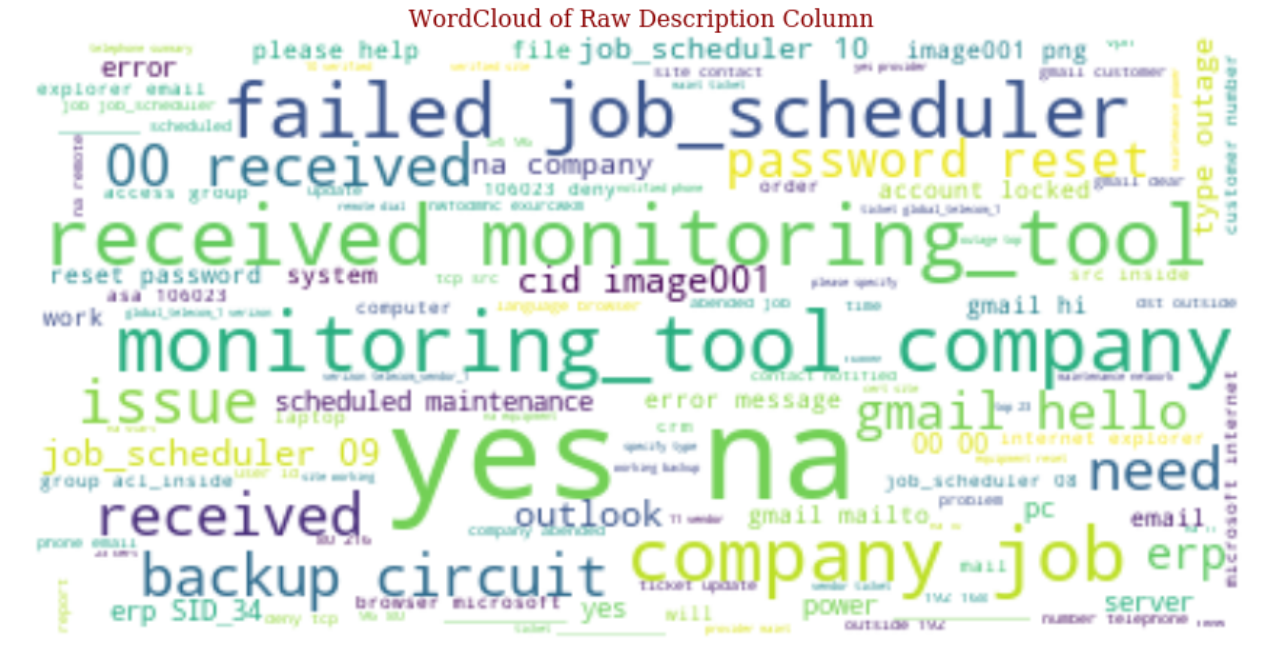
## Visualization

A TAG Cloud or Word Cloud is used to represent the Frequency of each word and hence we are checking how the Word cloud for each of the column looks before the data clean up

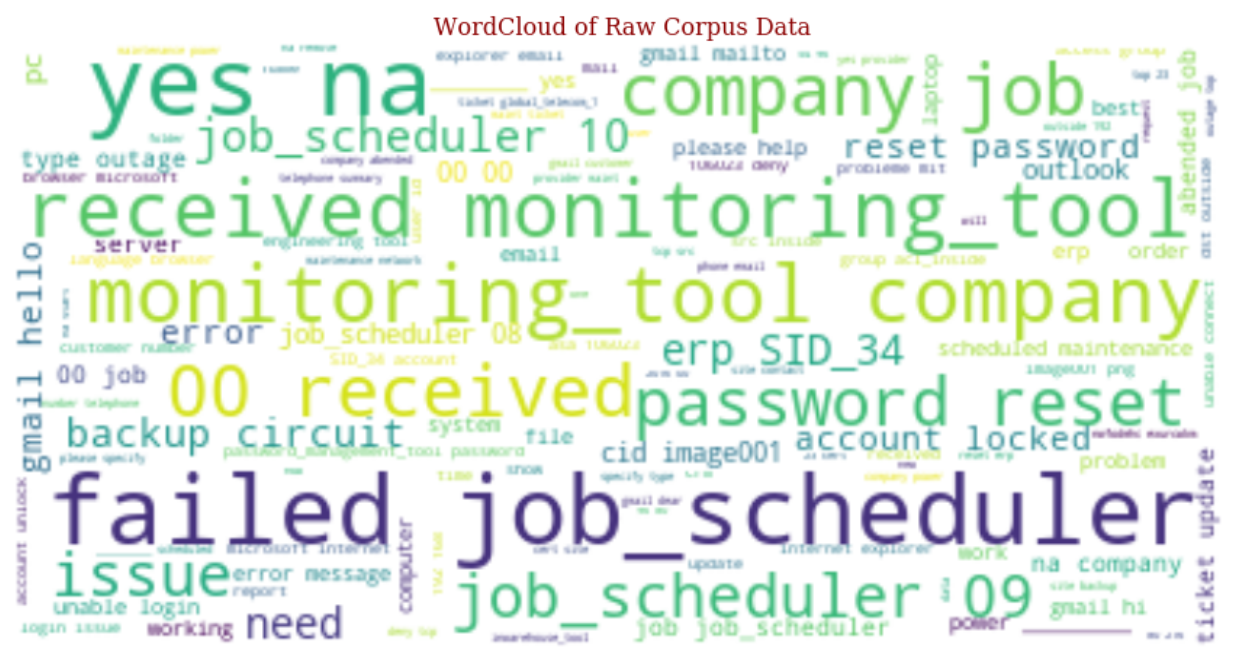
Word cloud for Title column



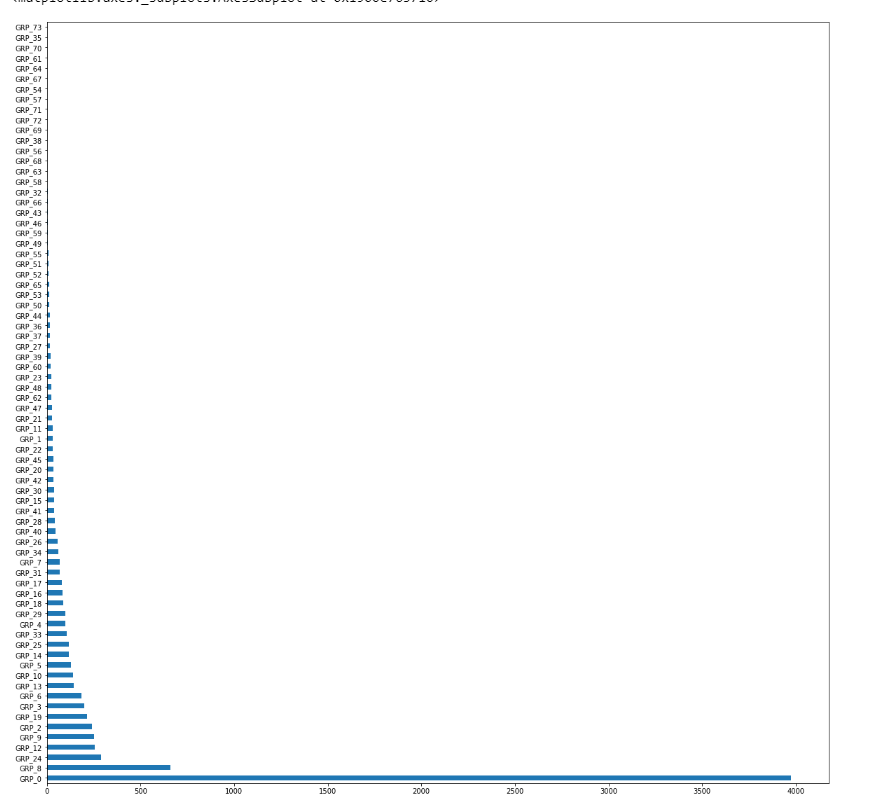
Word Cloud for “Description” Column



Word Cloud of Raw Data for the corpus - "FullDescription"



Data Distribution before cleanup



The data is now cleaned by converting all data into lower case and adding regex list as required to remove unnecessary text such as https:? , numbers, special characters, single alphabets, email tags and email ID’s , user names etc and store the cleaned text in 3 new columns called Rawdescription ,RawWordCount and CleanDescription

StopWords are the most common words in NLP for ex is, as the, a, an etc. For the purpose of analysing text data and building NLP models, these StopWords might not add much value to the meaning of the document.

For tasks like text classification, where the text is to be classified into different categories, StopWords are removed or excluded from the given text so that more focus can be given to those words which define the meaning of the text.

|  |  |
| --- | --- |
| **Sample Text with Stop words** | **Sample text without Stop words** |
| There is a book on the table | There book Table |

Here are a few key benefits of removing stopwords:

* Up on removing stopwords, dataset size decreases and the time to train the model also decreases
* Removing stopwords helps improve the performance as there are fewer and only meaningful tokens left. Thus, it increases classification accuracy

StopWords have been removed in the following steps

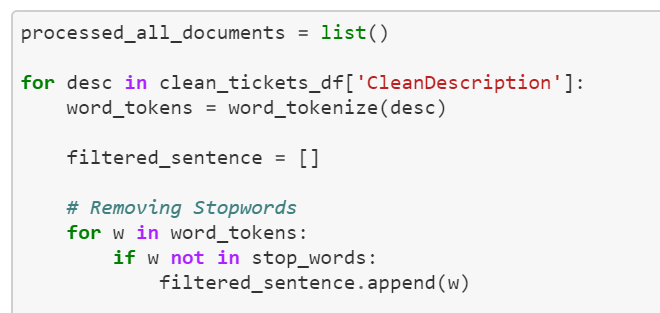
1. Text Classification
   * Spam Filtering
   * Language Classification
   * Genre Classification
2. Caption Generation
3. Auto-Tag Generation

We need to avoid removing StopWords in the below steps

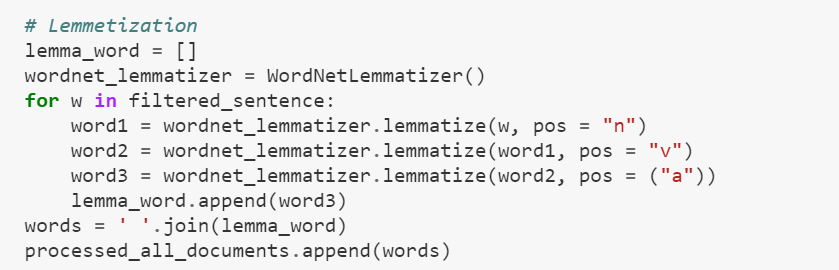
1. Machine Translation
2. Language Modelling
3. Text Summarization
4. Question-Answering problems

NLTK, or the Natural Language Toolkit, is a treasure trove of a library for text pre-processing. It’s one of my favourite Python libraries. **NLTK has a list of StopWords stored in 16 different languages.**

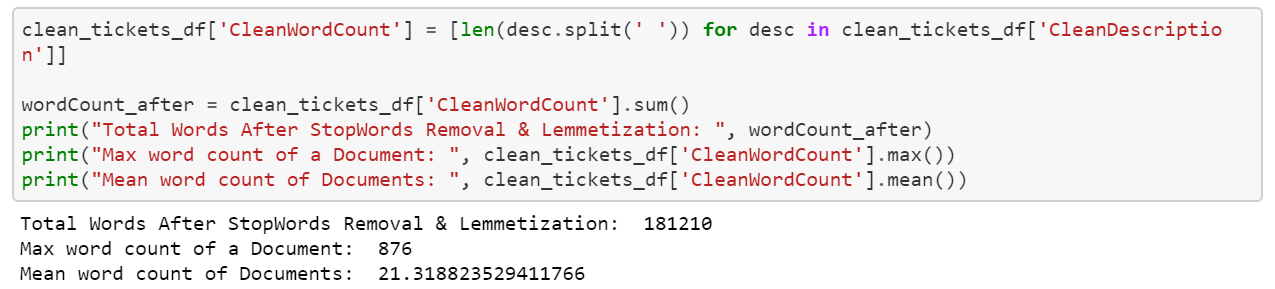
**Tokenization is a process of splitting up large body of texts into more smaller sentences or words. We tokenize the words using word\_tokenize function available as part of nltk.**



**Lemmatization** is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.  lemmatization does morphological analysis of the words.



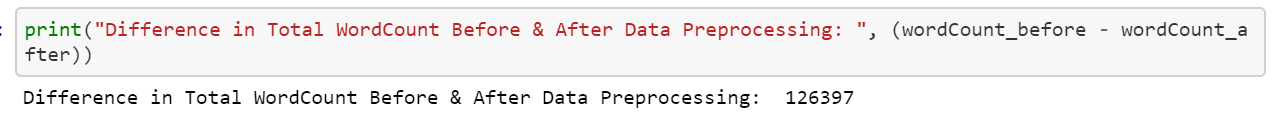
**After Tokenization and Lemmatization, it was found that**



We hae compared the data after StopWord removal and Lemmatization to the data before it . we can clearly see that the word count for each record has significantly decreased after the above process



Difference in the total word count before and after the StopWord removal process

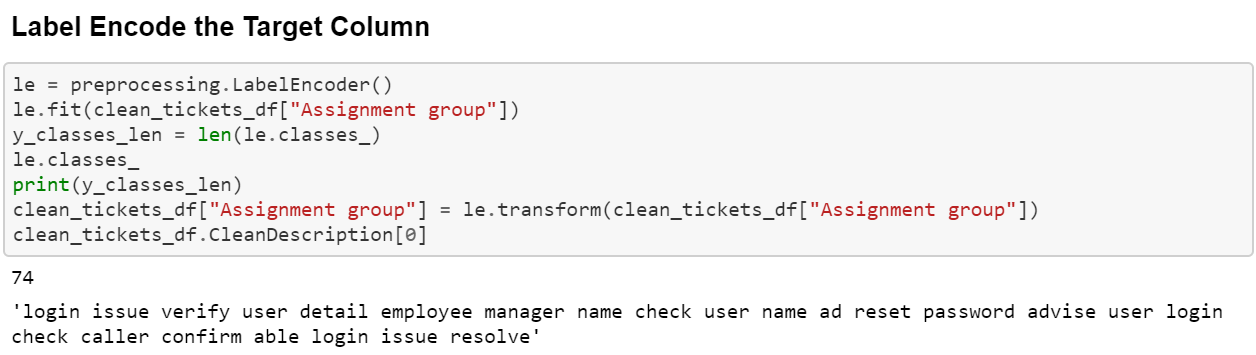


## LABEL ENCODING and BASIC MODELS

**Label Encoding** is an important pre-processing step for the structured dataset in supervised learning. It refers to converting the labels into numeric form so as to convert it into the machine-readable form.

**Example of Label Encoding**

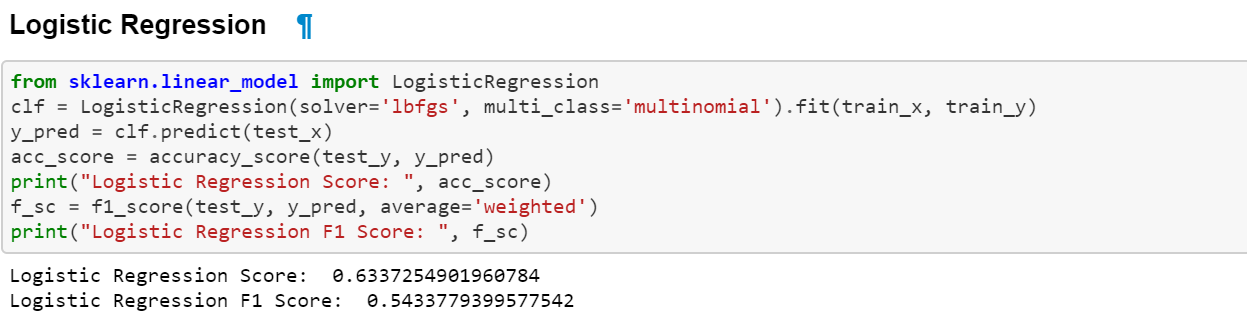
|  |  |
| --- | --- |
| Before applying Label Encoding | After applying Label Encoding |
| Tile | 0 |
| Description | 1 |
| Caller | 2 |
| Assignment\_Group | 3 |



## CREATING MACHINE LEARNING MODELS

Classification

Logistic Regression is known for its versatility and explain-ability. It is very simple to train and the results are interpretable as you can easily extract the most important coefficients from the model.



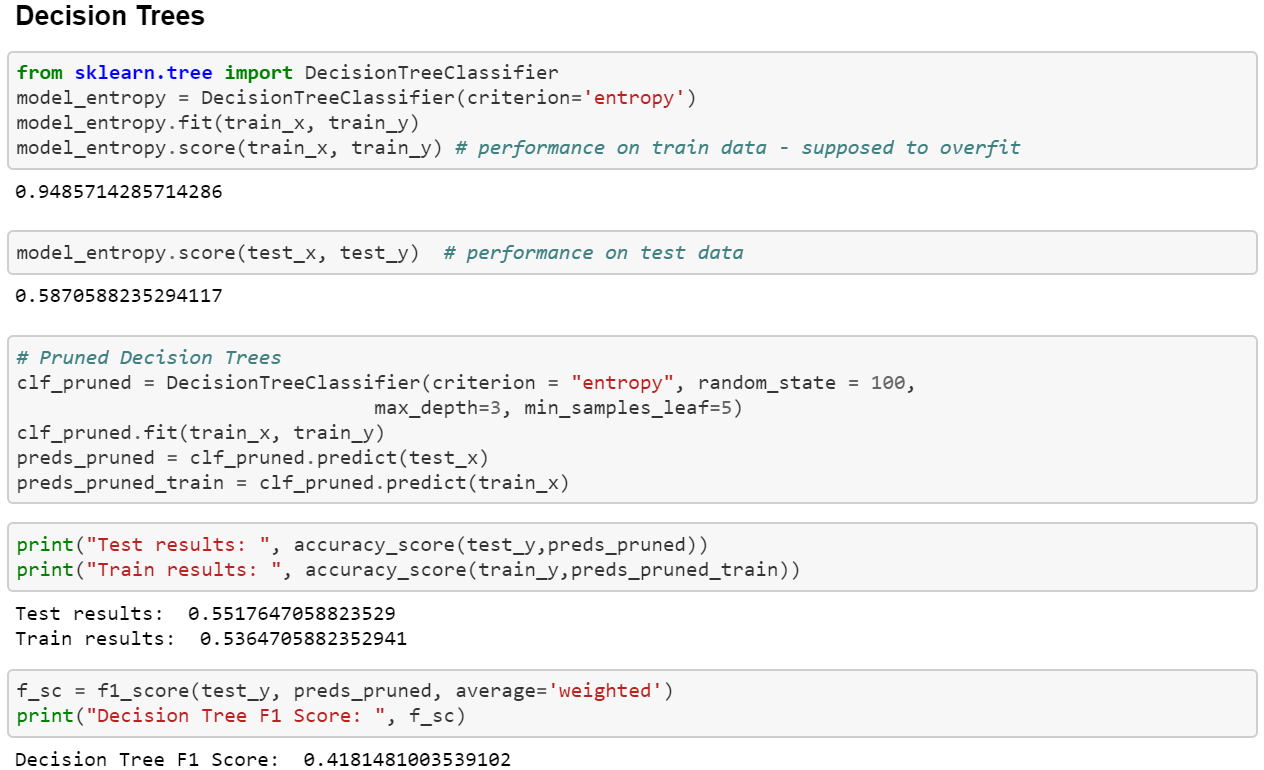
Naïve Bayes is a Set of classification algorithms based on **Bayes’ Theorem**.



Linear SVM



Decision tree



Ensemble Technique applied- Random Forests



Compare the mode score of all the above models



From the above table we can see that the accuracy of the above linear models is not as anticipated hence further Deep Learnings models need to be applied

Planned Activities

1. Building DL based models
2. Hyperparameters tuning
3. Any benchmarking if there
4. consolidation