Capstone Project – Final Report

Automatic Ticket Assignment (NLP)

Group - AIML19 Group 10A

Milestone – 2 Submission

07-06-2020

AIML June Group 10A Team Structure:

Mentor	:	**Amit Kayal**
Program Coordinator	:	**Divya Teresa**

Team Members:

- ✓ Hari Prasad Shanmugavelu
- ✓ Kaarthickeyan Palanisami
- ✓ Prasanna Chinnappan
- ✓ Sandesh Kirani Ravindra
- ✓ Shashikant Sangalad
- ✓ Shiva Kumar Chawla

Table of Contents

Abbreviations and Acronyms:

AIML	Artificial Intelligence Machine Language
EDA	Exploratory Data Analysis
IP	Internet Protocol
ITSM	Information Technology Service Management
IT	Information Technology
MTTR	Mean Time to Restore
NLP	Natural Language Processing
POS	Parts of Speech

Executive Summary:

This document is the final report of the Capstone Project - Automatic Ticket Assignment (NLP) for the Batch - AIML19Group10A. The Project is performed in the field of Natural Language Processing (NLP), an interdisciplinary field of Artificial Intelligence (AI), that uses many different techniques like Deep Learning to explore the interaction between human (natural) language and the computer language.

The Client partner is a large multinational organization with a setup of IT Service management processes supported by tools and Service Desk. Auto classification of incidents based on the previous assignment patterns, powered by NLP is the identified goal of this capstone project. Service Desks are the backbone of IT operations for addressing the requests and incidents of the Business/IT groups of any organization, as quoted below by one of the Service Desk Application Major.

Empower your service desk with this treasure trove of all things IT service management (ITSM). – ManageEngine.

"If we consistently exceed the expectations of employees, they will consistently exceed the expectations of our customers." – Shep Hyken

Intelligent Tickets assignment are becoming famous applications in the world of Service Desk now and some businesses are already taking advantage of this and developing Automatic Incident Assignment to complement their existing processes.

The ability to deliver IT services to employees in real-time while adhering to process workflows allows IT organizations to reduce MTTR and cost per ticket without sacrificing customer satisfaction or compromising proper IT governance and control.

- Rob Young in medium.com

The Automated Ticket assignment system will even result in rationalisation of the team structure of L1/L2 by enabling them in spending of their time and effort in some other useful work than the mundane assignment of tickets.

Introduction

Purpose:

The purpose of this Capstone project is to showcase the power of Natural Language Processing (NLP) and how they can complement the matured ITSM processes, especially the Service Desk of an organisation in automatically identifying the functional group to assign a service desk incident.

Objective:

To increase the service desk efficiency as well as the trust of the service desk users, by reducing both the time taken to assign the incidents to the right group and the error rates in incident assignment (to wrong groups).

Value Addition:

This will be a value addition which will be completely transparent to the service desk user community except for their satisfaction in their incidents getting addressed and issues resolved at lighting speed.

Technologies Used:

- **Python**: This application is developed using the Python. Natural Language Processing libraries are used extensively in this project.
- **Google Collaboratory** from Google Research, a hosted Jupyter notebook service is used for the python coding and running the code.
- **Github**, the software development platform has been used here for the collaboration of data, code, documentation.
- **Google Meet platform** is used for collaborating between the team members, for daily calls with screen sharing

1. Summary of problem statement, data and findings:

1.1 Summary of Problem Statement:

The problem in hand is the erroneous and time-consuming service desk incident assignments to the numerous functional groups of the service desk. The Organization is a multinational conglomerate with its presence across the world, with employees working in all the major time zone. It is also understood that users communicate using their native language also, with majority of communication in English.

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.

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.

High level assumptions considered regarding the data:

- Assignment Functional group will belong to L1/L2/L3 teams
- Assignment group is the target column (y)
- Incidents are not reassigned to the assignment group (there is no data about reassignment)
- No relation between incidents and no parent child between different incidents
- Data Assignment to Assignment group is assumed to be accurate as provided in dataset
- Caller is the person created the incidents

1.2 Data Set

The data file "input_data.xlsx" is provided as a prerequisite in the capstone project and this file contains the data set required as input to the project.

The data file (input_data.xlsx) contains four columns as described below,

Sl. No	Column Name	Description about the column	Type of Column
1	Short description	A short description of the incident, many a times a user written summary or highlight of the issue/request.	Free text column where user can write their own text.
2	Description	Lengthier description. Many a times a detailed and explained description, and a few times a copy of the short description.	Free text column where user can write their own text.
3	Caller	The user who raised the incident or the 'automated program' creating incidents.	Pulled from the user-id column of the registered users of service desk.
4	Assignment group	The Functional group, to which tickets were assigned eventually by the L1/L2 teams.	Selection (from a dropdown) by L1/L2 team after initial analysing

1.3 Data Findings:

- The names in the dataset are modified such that there is no relation can be established with the real names of any users.
- The Assignment Group column is the target variable and classes among which the incidents will be assigned
- The dataset has total of 8500 samples.
- The 8500 records are distributed across 73 Assignment groups
- Almost 50% of samples belong to one Assignment Group (Group-0) and this means data imbalance between the class
- Few of the Assignment groups 5 to 1 samples and these groups will impact the performance of the Model big time

- As depicted in the Visualisation-1 in section 5, the data is skewed and there is high imbalance between assignment groups (target classes).
- 3976 incidents were assigned to Assignment Group-0 and this is majority group
- There are two groups with only 1 record of data sample
- There are five groups with only 2 incidents assigned.
- There are five groups with only 3 incidents assigned to them.
- There are two groups with only 4 incidents assigned to them.
- There are only one group with 5 incidents assigned to them.

Implications:

- Data is highly imbalanced between Group-0 and Rest of the Assignment Groups
- Data imbalance will be impacting the Model performance and it will be biased towards majority classes

Findings about Short description and Description Columns:

- Short description in many incidents is a brief of the incident and the description (long) is the detailed write-up of the incident.
- In some incidents, short description and description are same.
- In some rows, Short description were blank, and, in few incidents, description was blank.
- There were some rows in which both the short description and description were same or having minor difference.
- In both short description and the description, there is date, time, IP addresses and special characters in the data in addition to multi-lingual data.
- There is around 10% of incidents with non-English characters.
- Data consists of IP addresses, which in some places are assumed to be supposed to be date.
- Approximately 900 incidents are related to failure of Job_scheduler.
- In these records, the text is same about failed jobs, except the jobid and the date and time, yet they are assigned to different groups
- Also, it was not possible to find a relation between job_id, data, time with the group assigned.

Implications:

- There is a lot of duplication, numbers, special characters. NLP does not handle numbers, special characters. These have to be removed / handled in pre-processing.
- As around 900 records have the same text, if the numbers (date, time and job_id) are removed, the train data might be insufficient to determine grouping for these records
- The non-English data have to be either handled with language translation or to be dropped

Findings in the Caller column:

- Majority of the Incidents raised by one particular caller were assigned to the same group. However, this is only an observation and the relation could not be established between caller and any other group.

Implications:

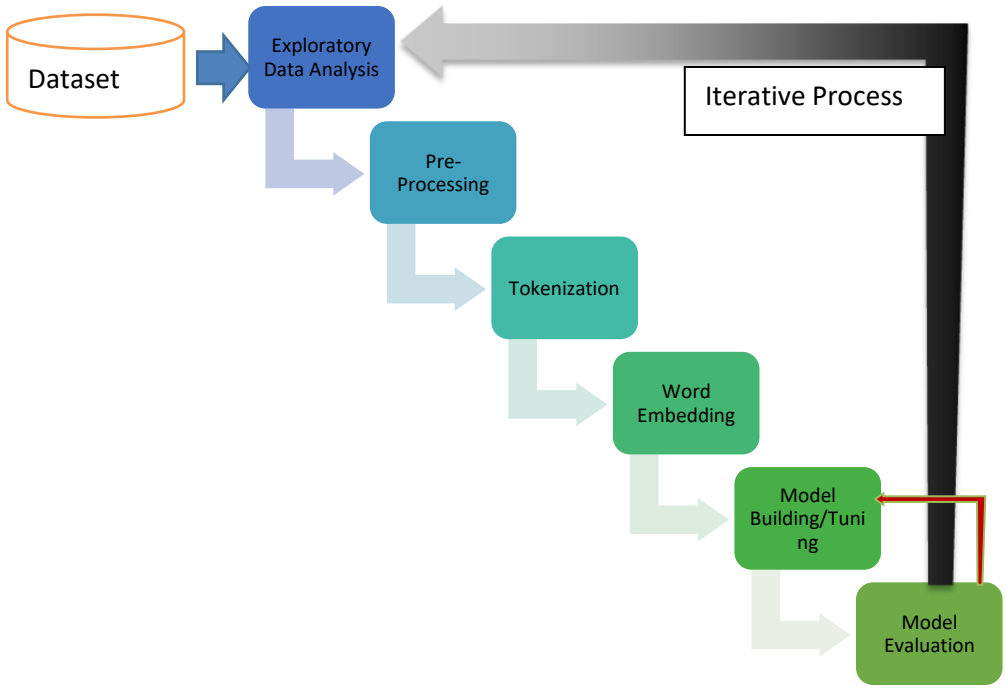
- The handling of caller column is briefed in Pre-processing section.

2. Overview of the Final Process:

2.1 Problem Methodology:

The problem statement involves text-based data as input and hence falls in the Natural Language Processing category of AI methodology. Following Flow diagram depicts the overall Pipeline and methodology followed end to end for addressing incident Ticket Assignment problem.

Predominantly there are SIX overall steps in the process and it's an iterative process where the there is always a room to the keep improving the performance of the Model.



NLP Pipeline Methodology

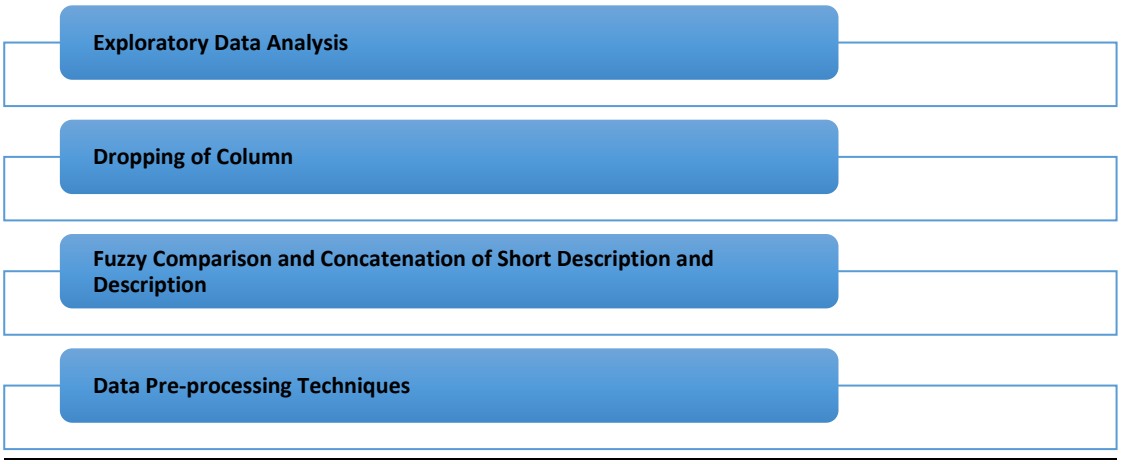
Salient Features of the Data,

- ✓ Data is highly imbalanced between Group-0 and rest of the Assignment Groups
- ✓ Short Description and Description columns will be making the Corpus of the Model
- ✓ Data seems to be a free text and does not have defined structure or format
- ✓ Similar text content in the incidents is assigned to multiple Assignment Groups in few cases
- ✓ One of the observations is that Large set of incidents are generated by same Caller

- ✓ The Callers are free to create incidents in any Language and hence we can find roughly 8 to 10% of incidents in languages other than English
- ✓ Large set of incidents are auto generated by the digital robot and do have same text content like Job Scheduler Id, Date, time stamp etc.

2.2 Data pre-processing steps:

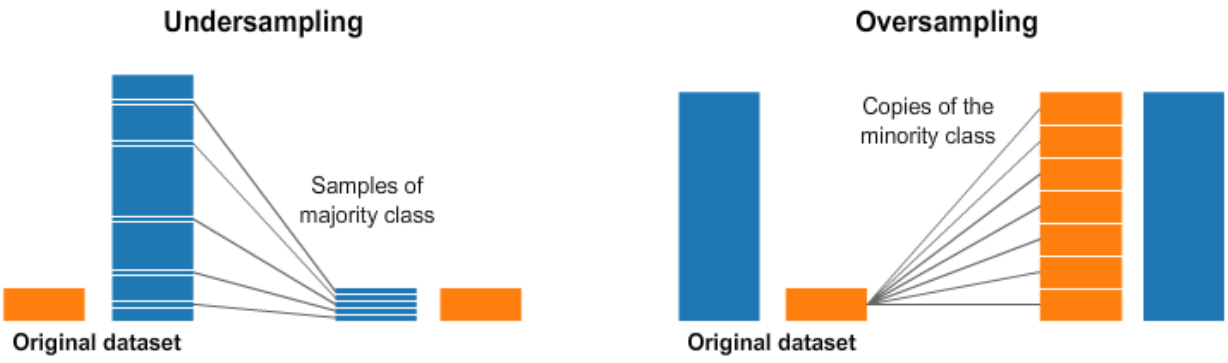
The following are the high-level data pre-processing steps followed in our solution,



Pre-Processing steps in detail,

Resampling

A widely adopted technique for dealing with highly unbalanced datasets is called resampling. It consists of removing samples from the majority class (under-sampling) and / or adding more examples from the minority class (over-sampling).



Despite the advantage of balancing classes, these techniques also have their weaknesses (there is no free lunch). 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.

1. Feature Engineering

Incident from one single caller assigned to multiple group. Hence, this data is biased

Refer to Visualizations section for further plots

bpctwhsn kzqsbmtp	810
GRP_1	6
GRP_10	60
GRP_12	8
GRP_13	4
GRP_14	1
GRP_18	3
GRP_29	1
GRP_44	1
GRP_45	7
GRP_47	2
GRP_5	96
GRP_57	1
GRP_6	89
GRP_60	16
GRP_8	362
GRP_9	153

As there is no relation between Caller column and the other columns, the caller column is dropped.

2. Comparison and concatenation of short description and description:

- As both short description and description have similar or same text in many cases, we used a package named fuzzywuzzy and its function to identify the fuzzy score (by comparing the words in two strings).
- If the two strings (short description and description) are same, the score will be very high, and a conditional statement is applied not to merge / concatenate the columns and take only one of the two column values. If the fuzzy score is less, then both the texts are concatenated.
- This is the method used to build the corpus (Combination of Short Description & description).

Sl. No	Pre-processing activity (in description and short description column)	Activity detail	Package
1.	Handling of missing Records	Dropping rows where either of them are null, as this will not happen in real world	Custom code
2.	Handling of special characters	All the special characters are removed from the corpus. Regular expression function .re is used for this purpose	Re Stopwords Lemmatization
3.	Handling of contractions text	Contractions is used for expanding and creating common English contractions in text. Eg. haven't - > have not.	Pycontractions
4.	Handling of URL, IP address, numbers	The URL, remove html code and IP addresses and numbers are removed from the concatenated column	BeautifulSoup
5.	Handling of multilingual text	The non-English records are identified in spreadsheet and removed from the dataset, as the non-English text is not taken as part of the corpus.	Detectlanguage
6.	Handling of whitespace/newline	The whitespaces/newlines are removed from the concatenated column	Custom code
7.	Handling of stopwords	Stopwords function was used to remove the stopwords from the corpus. Word with size (less than or equal to 2) to be discarded.	NLTK stopwords
8.	Identification of maxlen and minlen	The maxlen and minlen of the word corpus is identified by finding the length of each combined word (short description and description).	Custom code

Algorithms used,

The problem is a multi-class classification category and hence as part of the solution, we have tried with diverse Machine Learning Algorithms (classifiers) like below,

- Naïve Bayes (Multinomial) Classifier
- SVM (Support Vector Machine) Classifier
- KNN (K-Nearest Neighbours) Classifier
- Logistic Regression Classifier
- Ensemble Algorithm - Decision Trees
- Ensemble Algorithm - Gradient Boosting Classifier
- Word2Vec & Logistic Classifiers
- Linear Dense Networks
- LSTM & Bidirectional LSTM

Different techniques followed,

- ✓ Feature Engineering
- ✓ Fuzzywuzzy
- ✓ Sklearn utils resample for balancing the imbalance data
- ✓ Machine Translator
- ✓ Count Vectorizer
- ✓ TF-IDF
- ✓ Keras Pipeline to run different processes collectively
- ✓ pre-trained Embeddings using Glove
- ✓ Keras L1, L2 Regularizes
- ✓ Hyper parameters tuning and Call Back Functions like Early Stopping, Cyclic Learning Rate, Learning Rate Scheduler
- ✓ Used Libraries like BeautifulSoup to handle hyperlinks & URLs
- ✓ Handled Contractions using py-contractions library
- ✓ Ensemble Techniques and Voting Classifiers

3. Step-by-step walk through the solution:

3.1 Understanding the problem statement and the business domain

- ✓ As a team, we decided to have daily cadence calls to connect and brainstorm
- ✓ Used Git-hub for tracking our project
- ✓ Define plan with high level activities and due dates
- ✓ As a team, we did a deep dive into the problem statement by sharing each of our experience with incident management process
- ✓ Further analysed the give problem statement to define the broad category of the problem

Findings,

- ✓ This is very common problem among organization across industries
- ✓ There are multiple levels of parties involved till the ticket is assigned to the final assignment group
- ✓ Incidents are raised either by Service Help Desk personal or Digital Assistant Robot

3.2 Understanding the data set and Data Exploration (EDA)

- ✓ Used excel pivot and features to explore the data and relationships between various columns or features
- ✓ Compared Description & Short Description columns to make sure they are not repetitive and contain different text content
- ✓ Analysed the Caller Column to see if any inference with Assignment groups
- ✓ Looked for data in details for presence of URLs, Digits, special characters, new line characters, spaces
- ✓ Analysed the data for multiple language presence and ratio of the sample in English vs. other languages
- ✓ Used Google Sheets for identifying the languages
- ✓ We also used Fuzzywuzzy library to compare & assign a score to the short description and description columns before merging them together for larger corpus
- ✓ Used EDA's like value counts, Group by, various visualizations, word count, outlier with box plots, pie charts to explore the data

Findings,

- ✓ The data is very imbalance and majority of the samples are under Group-0

- ✓ Refer the Data findings section 2

3.3 Data Pre-processing

- ✓ We realized that data contains digits, special characters, URLs, new lines, spaces etc. and hence looked for different libraries to deal with these data findings
- ✓ Explored libraries like re, stop words from NLTK, lemmatization from NLTK, contractions by pycontractions, tokenizer from NLTK, stemming by NLTK, BeautifulSoup from bs4
- ✓ Explored google translator & goslate libraries for language translation
- ✓ Sklearn utils resample for data upsampling for minority groups
- ✓ We visualized the word clouds before and after data pre-processing to understand the high frequency words appearing in the corpus

3.4 Feature Engineering

- ✓ As we realized the both short description and description columns had different text and hence adding context to the corpus, so we decided to combine both columns before feeding to the model
- ✓ And hence we introduced a new column for combined description
- ✓ Introduced a column to document the comparison score, which is the output of fuzzywuzzy algorithm
- ✓ Had a column to capture the word counts for columns like description, short description and combined description
- ✓ We decided to drop Caller column as we could not find any inference with the assignment groups
- ✓ Before dropping Caller columns, we analysed used excel pivot to find the inference
- ✓ One of the observations was also that, large set of incidents were created by one Caller

3.5 Derive Train, test and Validation data split

- ✓ We went 80:20 ratios for splitting the data set into train & test data sets
- ✓ Used validation split of ration 0.2(20%) during model training (model.fit)
- ✓ Used standard test_train split library for splitting the data

3.6 Identify the relevant Algorithms

- ✓ Firstly, finalized that it's a text-based problem and hence decided to categorize under the NLP methodology

- ✓ Identified the given problem is a multi-class classification problem with presence of 74 classes in the data set
- ✓ Listed down various classifiers among supervised, unsupervised and neural networks which work better for text-based classification problem

3.7 Model Building & Model tuning

- ✓ Based on the above identified algorithms, we distributed within the group to try these algorithms
- ✓ We tested with various model architectures and techniques like count vectorizer, TF-IDF for supervised algorithms

Deciding Models:

- ✓ Automated ticket assignment to multiple groups between Group 0 to Group 74 and in a machine learning world, this is more of a Multiclass classification problem.
- ✓ Since our input data is textual in nature, and it falls under Natural Language Processing category, hence the following algorithms are used:

3.8 Model Fit & Evaluation

- ✓ We listed down various sklearn multi class classification evaluation metrics
- ✓ Confusion matrix using heatmap to visualize the true positives, false positives, true negatives and false negatives
- ✓ Classification report to analyse the class level, precision, recall, f1 score and support instances
- ✓ Macro and micro average accuracy
- ✓ Visualized model accuracy vs epoch and loss vs epoch
- ✓ Plotted accuracy vs learning rate visualization
- ✓ Trained the model using hyper parameters batch size, epochs for different combinations
- ✓ Trained the model using various call back functions to control the Learning Rate and used cyclic learning rate technique to improvise the performance
- ✓ Used hyper parameters like min_lr, patience, factor etc. as part of call backs
- ✓ Tuned the model with Kernel regularizer, recurrent regularizer and bias regularizer

3.9 Go back to Step 1 or 2 to improve the Model performance

- ✓ With initial model performance, we went back to the drawing board to analyse the data further

- ✓ Brainstormed on various Techniques to deal with data imbalance
- ✓ Discussed about various model architectures like Progressive Training, stacked models, re-sampling (up-sampling) and n-grams

4. Model Evaluation:

4.1 Performance of different models:

4.2 SVM (Support Vector Machine) Classifier

```
CPU times: user 4 µs, sys: 1 µs, total: 5 µs
Wall time: 8.11 µs

Test Accuracy  0.8203125

Train Accuracy 0.9632975260416666
```

accuracy			0.82	3072
macro avg	0.85	0.76	0.76	3072
weighted avg	0.82	0.82	0.79	3072

4.3 KNN (K-Nearest Neighbours) Classifier

```
CPU times: user 2 µs, sys: 0 ns, total: 2 µs
Wall time: 5.25 µs

Test Accuracy  0.66015625

Train Accuracy 0.6963704427083334
```

accuracy			0.66	3072
macro avg	0.58	0.41	0.45	3072
weighted avg	0.66	0.66	0.61	3072

4.4 Logistic Regression Classifier

```
CPU times: user 12 µs, sys: 2 µs, total: 14 µs
Wall time: 7.87 µs

Test Accuracy  0.8056640625

Train Accuracy 0.8355305989583334
```

accuracy			0.81	3072
macro avg	0.81	0.71	0.73	3072
weighted avg	0.79	0.81	0.77	3072

4.5 Naïve Bayes Classifier

```
CPU times: user 2 µs, sys: 0 ns, total: 2 µs
Wall time: 4.77 µs

Test Accuracy  0.603515625

Train Accuracy 0.6220703125
```

accuracy			0.60	3072
macro avg	0.46	0.27	0.31	3072
weighted avg	0.62	0.60	0.54	3072

4.6 Decision Trees

```
CPU times: user 2 µs, sys: 0 ns, total: 2 µs
Wall time: 5.01 µs

Test Accuracy  0.8323567708333334

Train Accuracy 0.9742838541666666
```

accuracy			0.83	3072
macro avg	0.83	0.81	0.82	3072
weighted avg	0.82	0.83	0.83	3072

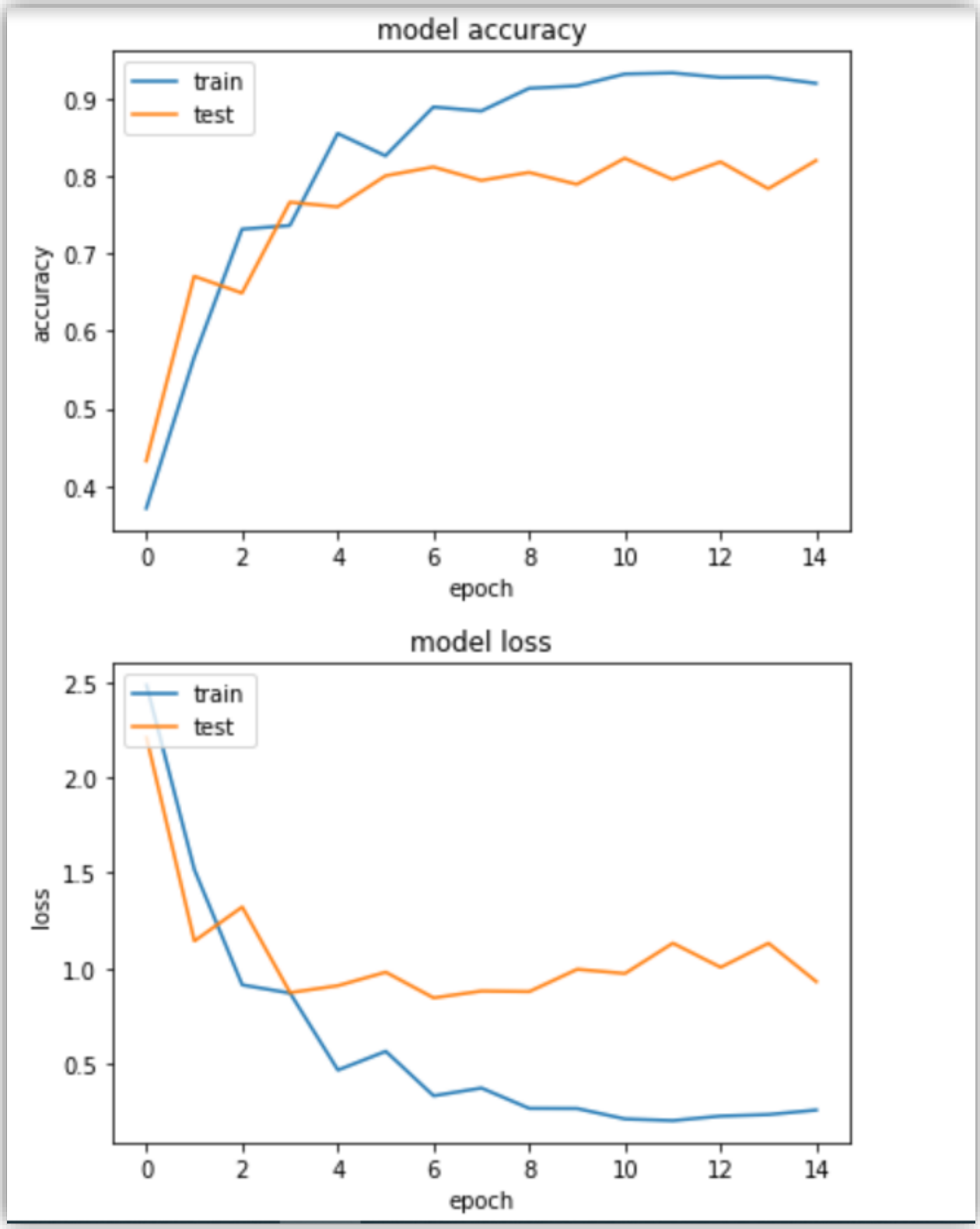
4.7 LSTM

```
# Final evaluation of the model

%time

score,acc = model.evaluate(X_test, y_test, verbose=1)
print("Score: %.2f" % (score))
print("Accuracy: %.2f%%" % (acc*100))
```

```
CPU times: user 3 µs, sys: 2 µs, total: 5 µs
Wall time: 9.54 µs
94/94 [==============================] - 17s 183ms/step - loss: 0.9287 - acc: 0.8202
Score: 0.93
Accuracy: 82.02%
```


model accuracy
train
test
accuracy
epoch
model loss
train
test
loss
epoch

Accuracy Report Matrix as presented in Interim Report

(Benchmark or Milestone 1):

Sl No	Algorithm	Accuracy
1.	SVM (Support Vector Machine) Classifier	**57.44%**
2.	KNN (K-Nearest Neighbours) Classifier	**56.74%**
3.	Logistic Regression Classifier	**62.10%**
4.	Naïve Bayes Classifier	**55.39%**
5.	LSTM	**62.50%**

Accuracy Report Matrix (Final Report or Milestone 2):

Sl No	Algorithm	Accuracy	Time (µs)
1.	SVM (Support Vector Machine) Classifier	**66.1**	**5.25**
2.	KNN (K-Nearest Neighbours) Classifier	**82.6**	**7.39**
3.	Logistic Regression Classifier	**81.0**	**9.3**
4.	Naïve Bayes Classifier	**60.4**	**4.77**
5.	Decision Trees	**83.1**	**6.91**
6.	LSTM	**82.3**	**6.44**

5. Comparison to benchmark

- ✓ Looking at the data and data distribution vs assignment groups, we considered 55% as benchmark as approx. 50% data samples are assigned to group-0 assignment group
- ✓ We improved the performance to 62% accuracy as part of the interim Report or Milestone 1
- ✓ We applied various techniques as described above, and we could achieve an accuracy of 84% in our final solution
- ✓ Yes, we very much improved upon our benchmark accuracy
- ✓ We worked data up-sampling techniques using sklearn utils resample library

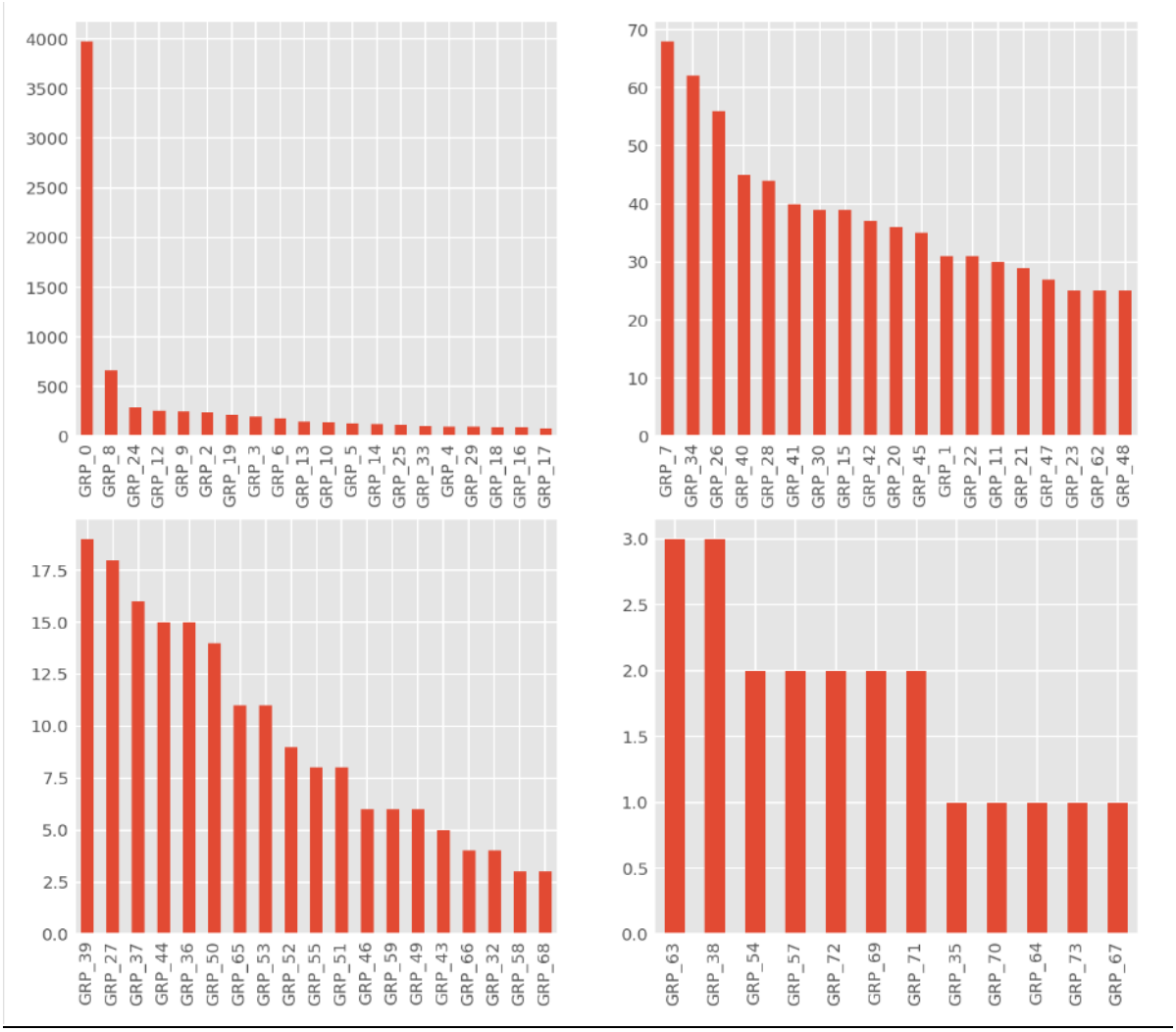
6. Visualizations:

Below are the visualisations for data samples distribution between 73 Assignment Groups:
Following four bar graphs are grouped as below,

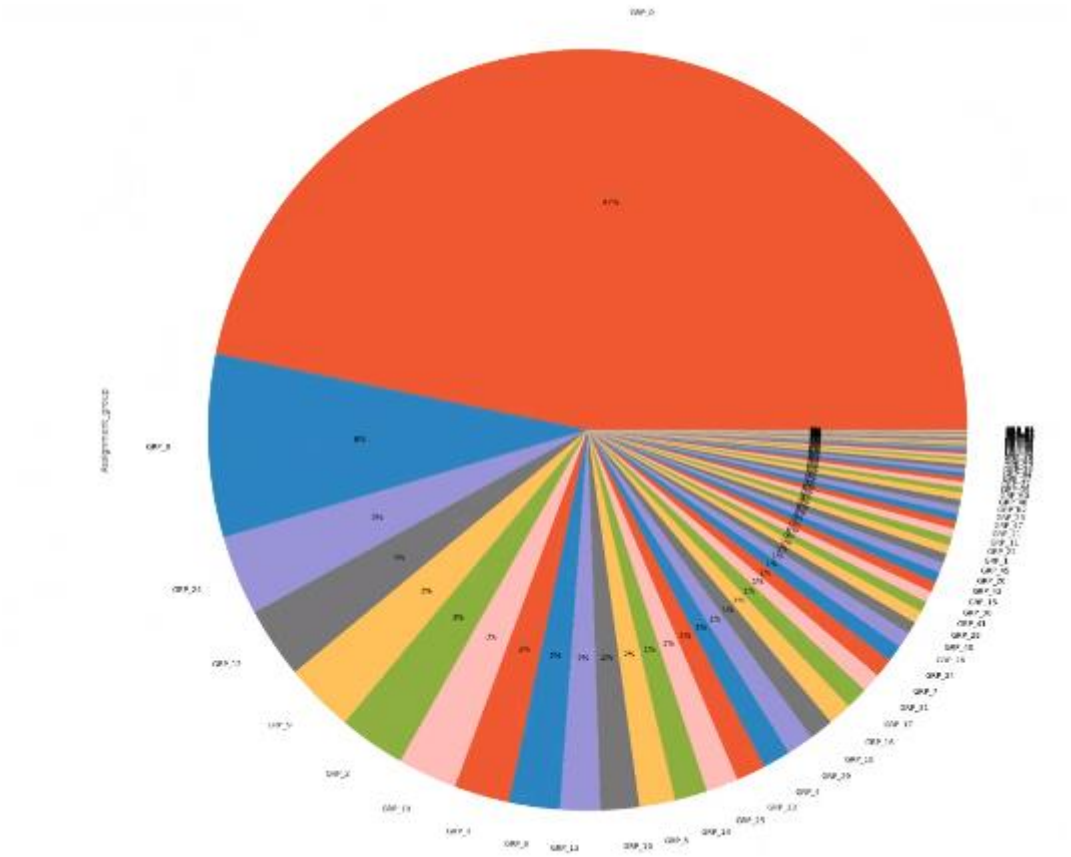
1. Top 5 Assignment groups in terms of sample count
2. Assignment groups between 6 and 30 in terms of sample count

3. Assignment groups between 31 and 50 in terms of sample count
4. Assignment groups between 51 and 73 in terms of sample count

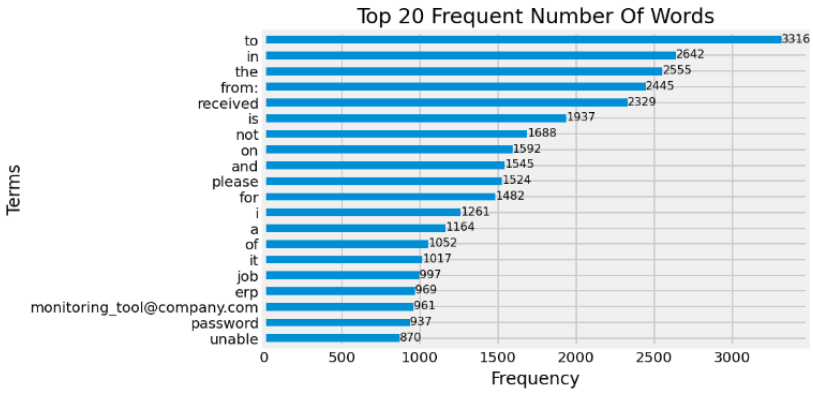
Visualization 1



Visualization 2 - Distribution of Incidents:



Visualization 2 - Frequent words before pre-processing:



Visualization 3 - Word cloud:

The Word counts before and after pre-processing were analysed.

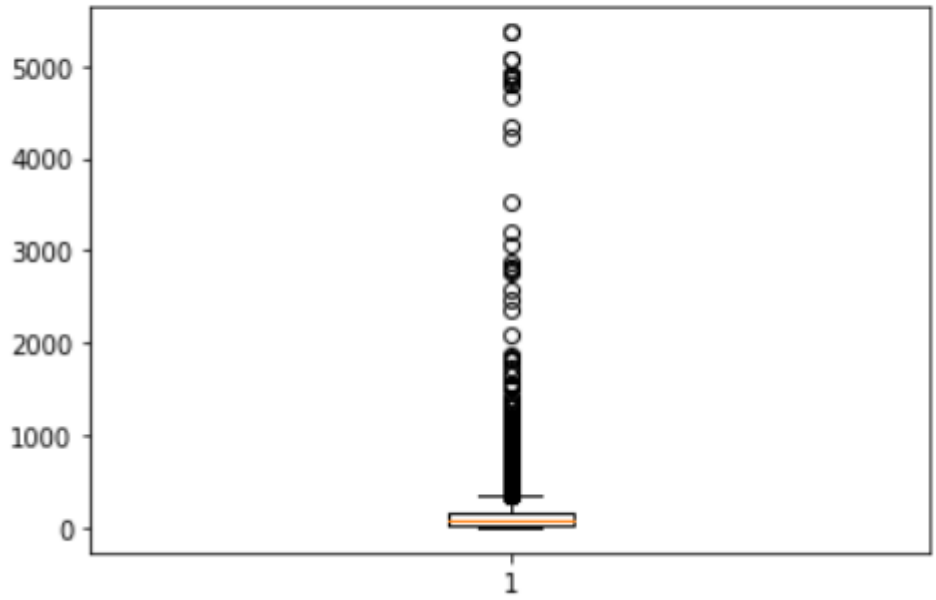
Word Count before Pre-Processing Work Count After Pre-processing

Word Count:

The word count of each incident was obtained and used in deriving the maxlen and minlen. For this purpose, a combined column of short description and description is used.

	Desc_1	Desc_1_count
4089	security incidents - (sw #in33895560) : mage...	981
4087	security incidents - (sw #in33895560) : mage...	980
5433	security incidents - (#in33765965) : possibl...	960
8002	security incidents - (sw #in33544563) : poss...	838
7997	security incidents - (sw #in33544563) : poss...	838
7995	security incidents - (dsw #in33407676) : tra...	731
7989	security incidents - (dsw #in33407676) : tra...	731
3965	security incidents - (#in33809307) : possibl...	720
7345	security incidents - (sw #in33501789) : broa...	705
3530	security incidents - (#in33944691) : possibl...	527
7984	security incidents - (dsw #in33390850) : sus...	440
7982	security incidents - (dsw #in33390850) : sus...	440
3706	security incidents - (#in33924718) : possibl...	416

```
Combined_Description length:
Mean 140.12 words (265.774747)
```



7. Implications

- ✓ The final solution will reduce the time spent by L1 or L2 teams to re-route the incidents to proper assignment groups automatically
- ✓ The solution will help to save the number of hours spent by resources on the manual activities of re-assigning to the appropriate group and hence the saved time can be spent on other productive activities
- ✓ Minimizes the Human errors in identifies the appropriate groups
- ✓ Time saved is directly proportional to the dollars saved and hence better operating margin
- ✓ Also inspires the organizations to utilise the AI technology for various other such human driven activities

8. Limitations

- ✓ Currently our Model has limitation to handle multi language texts

- ✓ It is not production ready because as we have not tested by deploying it on various other platforms and testing for different kinds of data inputs

9. Closing Reflections

- ✓ This project helped us understand doing solution and structuring for a machine learning problem
- ✓ We learnt a lot on how to handle a real time business problem end to end
- ✓ This project gave us a quite a confidence in handling the real time business problems
- ✓ Made us to explore various machine learning techniques and NLP methodology in detail
- ✓ We will try advanced topics like Bert, albert etc. and also Stanford libraries next time
- ✓ Helped us to learn from the each other in the group and also from mentor on different aspects of the solution

Improvements Areas

The following areas of improvement will be taken up once the current work is completed ahead of time.

1. **Handling multi-lingual data:** By identifying the non-English data and using goslate method
2. **Usage of POS**
3. **Flask:** To use Flask and exposing model through API, where we can use the API in any ITIL tool and test the model
4. **Stanford NLP usage:**
5. **Adding test data:** Adding test data manually into dataset, wherever needed
6. **Retrain the model based on user feedback (Reassignment happens):** i.e. if model does wrong assignments / reassigning happens then we take that data from business / client and we will try to retrain the model to do the proper assignment
   IP Address: Finding meaningful data from IP address by finding the country of IP

- Progressive Training:
  - Initially train the model for majority classes
  - Pass the learning rates from the learnt model to the next model for minority classes
  - Models can be multiple for a section of group, depending on the data samples
- Network Architecture Tuning:
  - Bring in Batch Normalization and additional Dense layers with RELU function
  - Try Convolution 1D and special dropout
  - Try Bert's algorithm
- Data Up sampling and Down sampling
- Class weights by normalising between 0 and 1
  - Penalise classes only which are under performing
- Negative mining
  - Analyse the class level performance
  - Deep dive into the classes where model is not performing well
  - Fine tune the model and define the strategies accordingly
- Try Hyper Parameter Tuning like Cyclic Learning Rate

10. Appendices:

10.1 References:

- https://www.tensorflow.org/
- https://keras.io/
- https://towardsdatascience.com/
- https://elitedatascience.com/
- https://machinelearningmastery.com/
- https://www.greatlearning.in/
- https://medium.com/
- https://stackoverflow.com/
- https://www.kaggle.com/
- https://github.com/
- https://pypi.org/

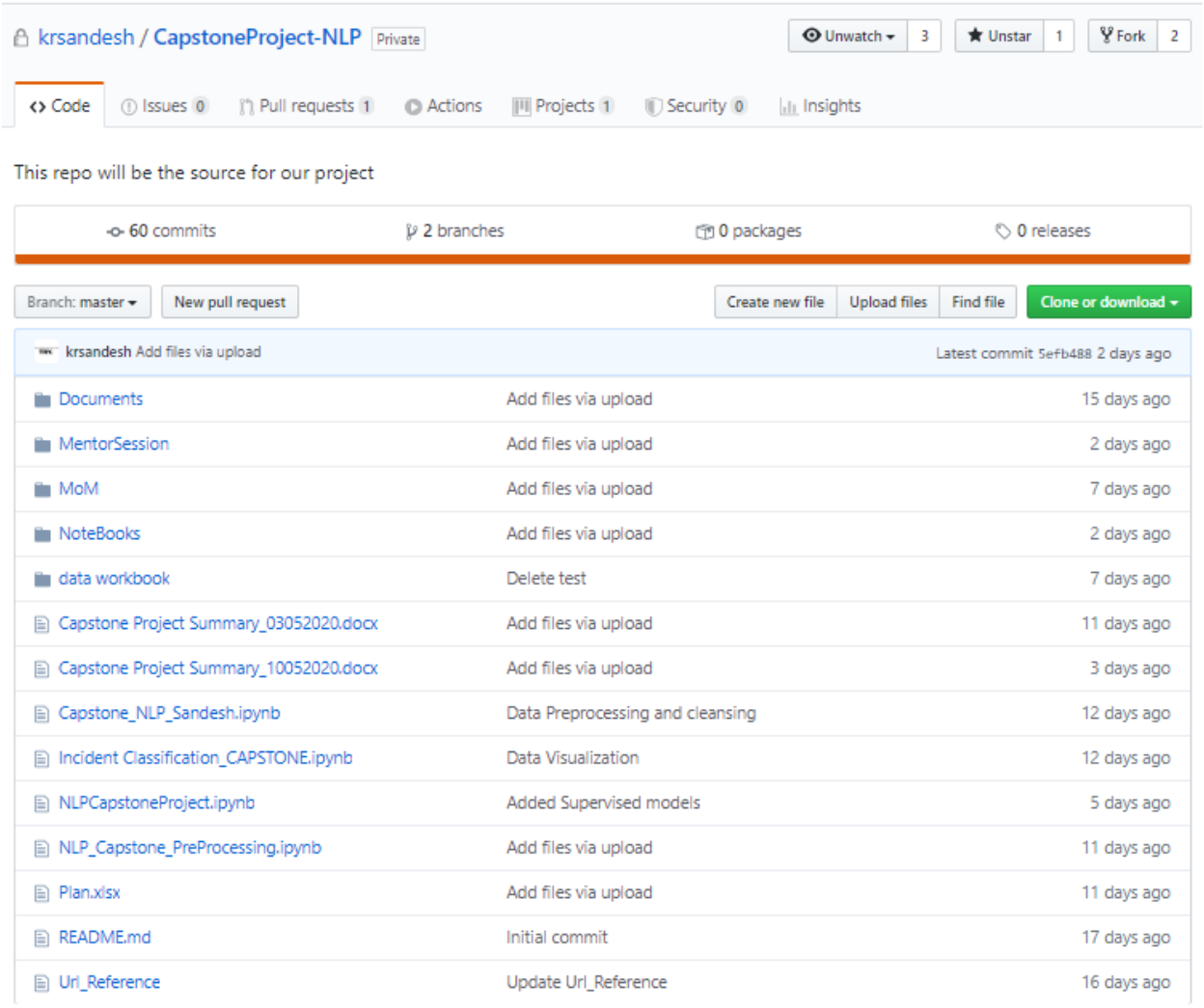
10.2 Appendix 2 Libraries used:

Sl No	Library/Function	Usage
1.	import matplotlib.pyplot as plt	For creating the visualization as part of EDA
2.	import seaborn as sns	For visualising (plotting) the data findings
3.	import os	This Python's standard utility module is used to interact with this file system, in our case for mounting the google drive and changing the path to working directory
4.	import nltk	Natural Language Toolkit provides easy-to-use interface to work with human language data. There are a suite of libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers
5.	nltk.download('stopwords')	This is used for identifying and removing the stop words from the corpus
6.	nltk.download('wordnet')	This large word database is used to identify and remove the of English Nouns, Adjectives, Adverbs and Verbs from the corpus

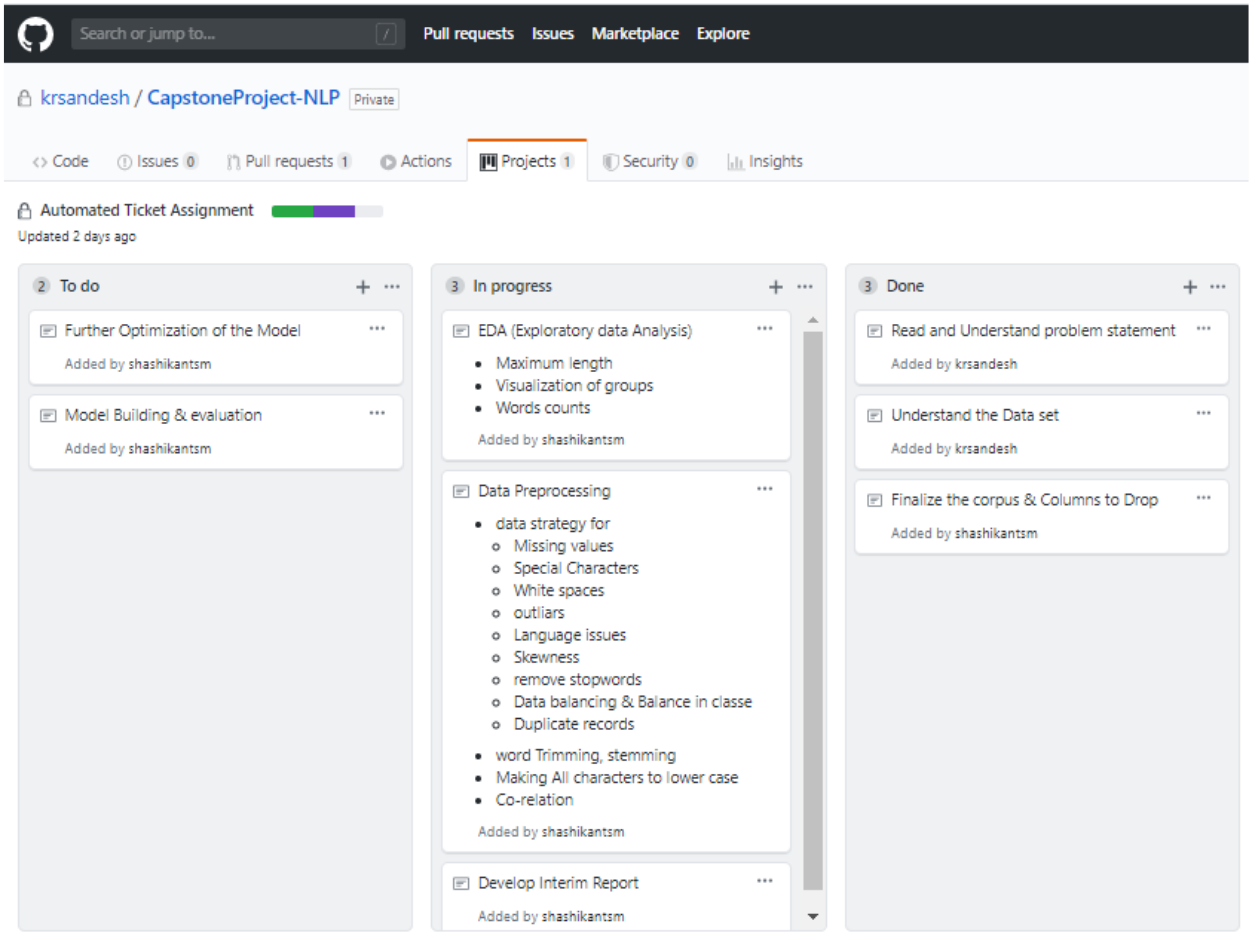
Sl No	Library/Function	Usage
7.	nltk.download('punkt')	This is used to tokenize the sentence
8.	import re	Regex module provides support for regular expressions like search and match
9.	fuzzywuzzy	Fuzzy string matching uses Levenshtein Distance to calculate the differences between sequences in a simple-to-use package. This function is used to identify similarity between short description and description, so that a decision whether to concatenate or take one of the columns
10	Contractions	This function is used to identify and fix contractions such as `you're` to you `are`

10.3 Appendix 3 Github usage:

Github, the software development platform has been used here for the collaboration of data, code, documentation. A private project was created and all the team members have been given access to the project. The project code, project inputs, data and the minutes of meeting, both internal and with the mentor are stored and maintained in the repository.



The tasks assigned to the team is also maintained and managed in the Github as below:



Issues faced in Github:

The document and spreadsheets can't be edited in Github. They have to be downloaded, edited and uploaded again.