qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| AUTOMATIC TICKET ASSIGNMENT  AIML Online Capstone Group 6  05-Jul-20  Group 6 |

# Summary of problem statement, data and findings

## Problem Statement

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.

## Abstract

Applying traditional machine learning and neural network-based NLP to automatically classify tickets and assign them to the right owner in a timely manner to save effort, increase user satisfaction and improve throughput in the ticketing pipeline of an organization.

## Data & Findings

1. The dataset comprises of **8500 rows** and **4 columns**
2. All columns are of type object containing textual information.
3. There are **8 null/missing values** present in the Short description and **1 null/missing values** present in the description column
4. **Password reset** is one of the most occuring tickets which reflects in the Short description column.
5. The top occuring Description in the dataset is only the text **'the'**, which absolutely doesn't make any sense. hence by looking at the Short description of such rows reveals that these are also a category of Password reset.

### Data provided in format

XLSX/CSV

### Total Records

8500

### Data Fields

|  |  |
| --- | --- |
| Short description | A summary of the issue faced by the user |
| Description | Detailed description of the issue |
| Assignment group | GRP\_0 ~ GRP\_73 (total 74 classes of Assignment group) |

### Sample data

| **Short description** | **Description** | **Assignment group** |
| --- | --- | --- |
| login issue | -verified user details.(employee# & manager na... | GRP\_0 |
| outlook | \r\n\r\nreceived from: hmjdrvpb.komuaywn@gmail... | GRP\_0 |
| cant log in to vpn | \r\n\r\nreceived from: eylqgodm.ybqkwiam@gmail... | GRP\_0 |

### Distribution of classes and Observation

1. High imbalance seen in data for target column in our dataset with GRP\_0 having highest percent of representation
2. Many classes with very little representation.
3. Null values in Data:
   1. Short description **8**
   2. Description **1**
   3. Assignment group **0**
4. Observed few Ticket having Non-English ticket descriptions

# Overview of the final process

1. Loaded the given dataset
2. Performed EDA on the dataset to identify the
   1. Distribution of data,
   2. Languages,
   3. Length of each column of dataset.
3. Perform pre-processing
   1. Replace email Ids
   2. Contractions
   3. Lemmatization
   4. Replace the gibberish text using FTFY
   5. Detect the language
   6. Translate the non English content to English using Azure API
4. Identify the deterministic rules using MS SQL and MS Excel
5. Apply Deterministic rules on Dataset to address the groups before modelling
6. Perform text summarisation
7. Perform Up sampling on complete dataset except Group 0.
8. Perform Clustering on Group 0 records alone
9. Combine both the Up sampled and clustered data frames
10. Define independent and dependent features
11. Perform Label Encoding on dependant feature (target column)
12. Perform Vectorisation on independent features using TF-IDF/Tokenizer
13. Split the data to train and test dataset in 70:30 ratio
14. Apply Machine Learning models
    1. Support Vector Machine (SVM)
    2. Naïve Bayes Classifier
    3. Random Forest
    4. Light GBM
15. Apply Deep learning models
    1. LSTM
    2. Bi directional LSTM
    3. RNN
    4. Attention
16. Identify the best model based on the predictions

# Step-by-step walk through the solution

## Load Dataset

Loaded the input csv file into pandas data frame

## EDA on dataset

1. Distribution of data
2. Identified the languages
3. Most common words
4. Top n words
5. Bi grams
6. Trigrams

## Pre-Processing

1. Replace email Ids – Email Ids of users are replaced with common text ‘Email Address’ from Description and Short Description as it does not hold any significance.
2. Expanded all contractions as this is an important step, because it will reduce disambiguation between similar phrases
3. Grouped together the different inflected forms of a word using NLTK Lemmatization so they can be analysed as a single item
4. Used Regex Library for removal of
   1. Trailing spaces,
   2. line breaks and tabs (\r\n\t),
   3. Special characters and
   4. Extra spaces.
   5. Convert the data having garbled text such as mojibake using Ftfy Library.

## Foreign language detection and translation

1. Identified around 28 languages - For language identification, we used Facebooks's fastText library which can recognize more than 170 languages and classify thousands of documents per second.
2. The model returns back two tuples back. ISO code and the confidence level. # ([['\_label\_de']], array([[0.96568173]]))
3. For converting the ISO code to language name we used pycountry library. eg: de --> German
4. Translated using
   1. Goslate
   2. Google Translate API and
   3. Azure APIs
5. Identified better translation accuracy using Azure translator (translate 3.5.0) which uses Microsoft Translation API, a cloud-based machine translation service that extends the reach of apps in more than 60 languages

## Deterministic Rules

Deterministic rules are fixed set of rules /conditions that provide an accurate match on given dataset.

The dataset provided is having 74 groups and the data distribution is not balanced. The likelihood of predicting the accurate group may vary using a ML model due to this imbalance and may cause over fitting.

To eliminate this problem, we have selected the classes with less than 20 samples and tried to identify the similarities to accurately identify each of these cclasses.

We have queried the data for similarities within each class by loading data into Excel and SQL. We were succesfully able to identify rules to predict 12 classes completely and 4 classes partially.

The deterministic rules are applied on the translated data set. The predicted samples are assigned a 'predicted group' as a new column in dataset.

## Text Summarisation

Summarization aims to highlight important information within a large corpus. We have used Gensim summarization on our dataset

## Up sampling

Imbalanced datasets are those where there is a severe skew in the class distribution, such as 1:100 or 1:1000 examples in the minority class to the majority class.

This bias in the dataset can influence many machine learning algorithms, leading some to ignore the minority class entirely. This is a problem as it is typically the minority class on which predictions are most important.

One approach to addressing the problem of class imbalance is to randomly resample the dataset. The two main approaches to randomly re sampling an imbalanced dataset are

* 1. Deleting samples from the majority class known as under sampling, and
  2. Duplicating samples from the minority class, known as Over sampling.

In our case, we are not considering under sampling as it would result in loss of data. Instead we have used up sampling using Random oversampling technique.

We are passing the cleaned data set excluding the majority class (Group 0) to this over sampler. We were able to up sample 4740 minority class records (input) to 40077 records (output).

## Clustering

We have clustered the records under Grp\_0 using k -means clustering. This is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

## Define independent and dependent features

We are concatenating the Short description and Description as complete description and Assignment Group as Target.

Complete Description is considered as Independent attribute and Target (Assignment Group) is Dependent attribute.

## Label Encoding

Labels in Target (Dependent) column are encoded using sklearn Label Encoder. SkLearn Label Encoder encodes target labels with value between 0 and n\_classes-1.

## Vectorization

Vectorization is used to score the relative importance of words. This can be done by either using TF-IDF or Keras Tokenizer.

### TF-IDF

Term Frequency (TF) is the number of times a word appears in a document divded by the total number of words in the document.

Inverse Data Frequency (IDF) is the log of the number of documents divided by the number of documents that contain the word w. Inverse data frequency determines the weight of rare words across all documents in the corpus

TF-IDF is the TF multiplied by IDF.

### Tokenizer

This class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf-idf.

## Splitting Datasets

We have used the train\_test\_split function for splitting a single dataset into training and testing in 70:30 ratios.

The testing subset is for building the model. The testing subset is for using the model on unknown data to evaluate the performance of the model.

## Machine Learning Models

### Support Vector Machine (SVM)

The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

### Naïve Bayes Classifier

A naive Bayes classifier considers every feature to contribute independently to the probability irrespective of the correlations.

### Random Forest

Random forests are an ensemble method in machine learning that combines the wisdom of many different decision trees. By choosing the majority opinion from among all the decision trees in their collection, random forests can improve their performance and accuracy.

### Light GBM

LightGBM uses histogram-based algorithms which helps in speeding up training as well as reduces memory usage. This algorithm constructs trees leaf-wise in a best-first order due to which there is a tendency to achieve lower loss.

## Deep Learning Models

### LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

### Bi-Directional LSTM

Bidirectional RNN (BRNN) duplicates the RNN processing chain so that inputs are processed in both forward and reverse time order. This allows a BRNN to look at future context as well.

# Model evaluation

Describe the final model in detail. What was the objective, what parameters were prominent, and how did you evaluate the success of your models?

# Comparison to benchmark

How does your final solution compare to the benchmark you laid out at the outset? Did you improve on the benchmark? Why or why not?

# Visualizations

In addition to quantifying your model and the solution, please include all relevant visualizations that support the ideas/insights that you gleaned from the data.

# Implications

How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?

# Limitations

What are the limitations of your solution? Where does your model fall short in the real world? What can you do to enhance the solution?

# Closing Reflections

What have you learned from the process? What you do differently next time?

# Flow Chart

Read the dataset

Identify the different languages available in dataset

Translate the description and short description to English

Replace any email addresses in Description / Short Description with common text ‘Email Address’

Up sample the groups other than GRP\_0

Apply text summarization on filtered dataset

Identify the records which are not predicted for further processing.

Apply the deterministic rules on given dataset to predict the Assignment group.

Identify the deterministic rules on given dataset to predict the Assignment group.

Identify the deterministic rules to directly identify the groups.

Apply clustering on Grp\_0

Concatenate upsampled and clustered dataframes

Concatenate SD and Description as complete description

Apply tokenisation using TF IDF on independant column(s)

Split the data into train and test datasets in 70:30 ratio.

Apply ML models – SVM, Naive Bias, LightGBM

NN models-

Tokenise using keras tokenizer

Apply LSTM, Bi LSTM, Attention, GRU,

Evaluate the accuracy of each of the models to identify the best model.