Capstone Project AIML

AUTOMATIC TICKET ASSIGNMENT

Interim Report

- Executive Summary
- As is Process Map
- To-be Process Map
- Approach to be taken
- Exploratory Data Analysis
- Pre-processing
- Model Building and Performance
- Way Forward

Project Team : Meenakshi Awasthi, Deenadayalan, Geethika K, Bhasha Kapoor, Hemanth Kumar S N

Executive Summary

Summary of problem statement, data and findings

Problem Statement

- The assignment of incidents to appropriate IT groups is a manual process, Manual assignment of incidents is time consuming and requires human efforts. Due to human intervention, errors and resource consumption is carried out ineffectively because of the misaddressing.
- 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).
- 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

Goal

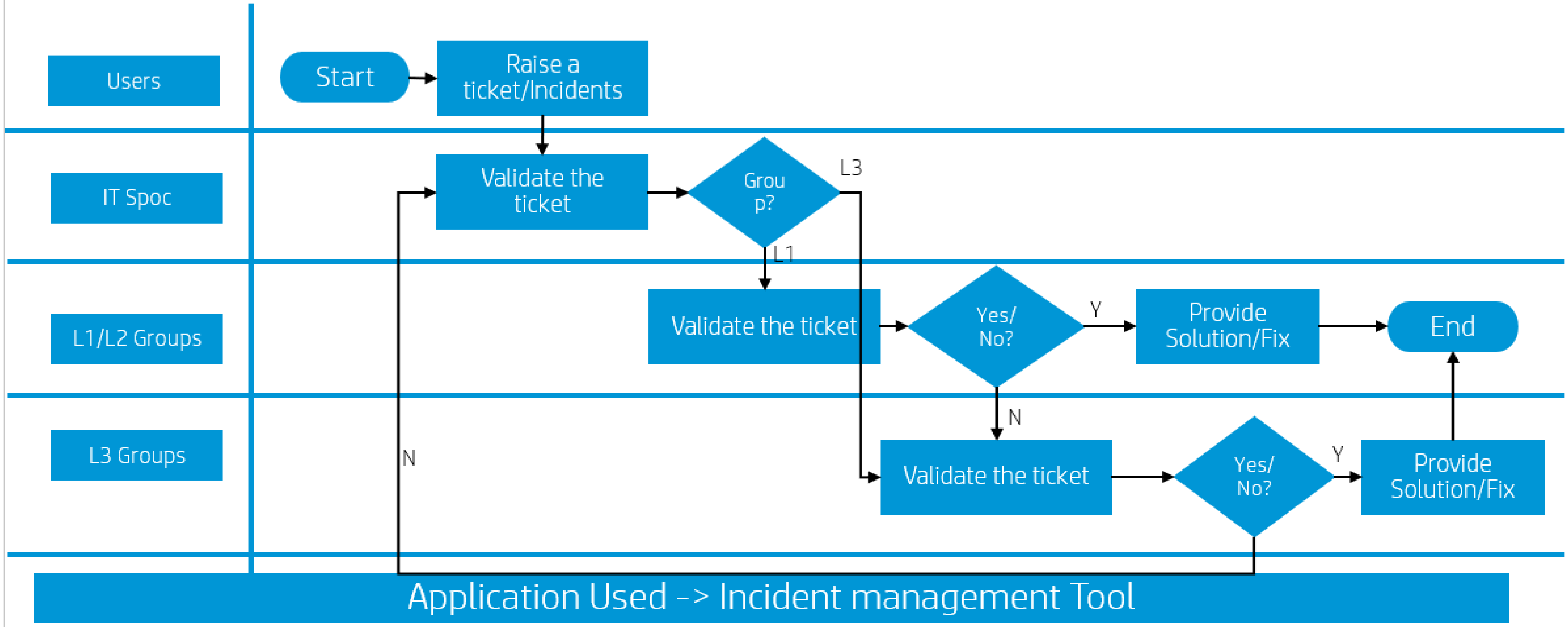
- Goal is the develop or build a classifier that can classify the tickets or incidents accurately to the right team/group by understanding and analysing the text given in the dataset based on the past incidents raised.

Benefits

- Automating of ticket assignment will reduce assignment error and time spent to review SOP for assignment

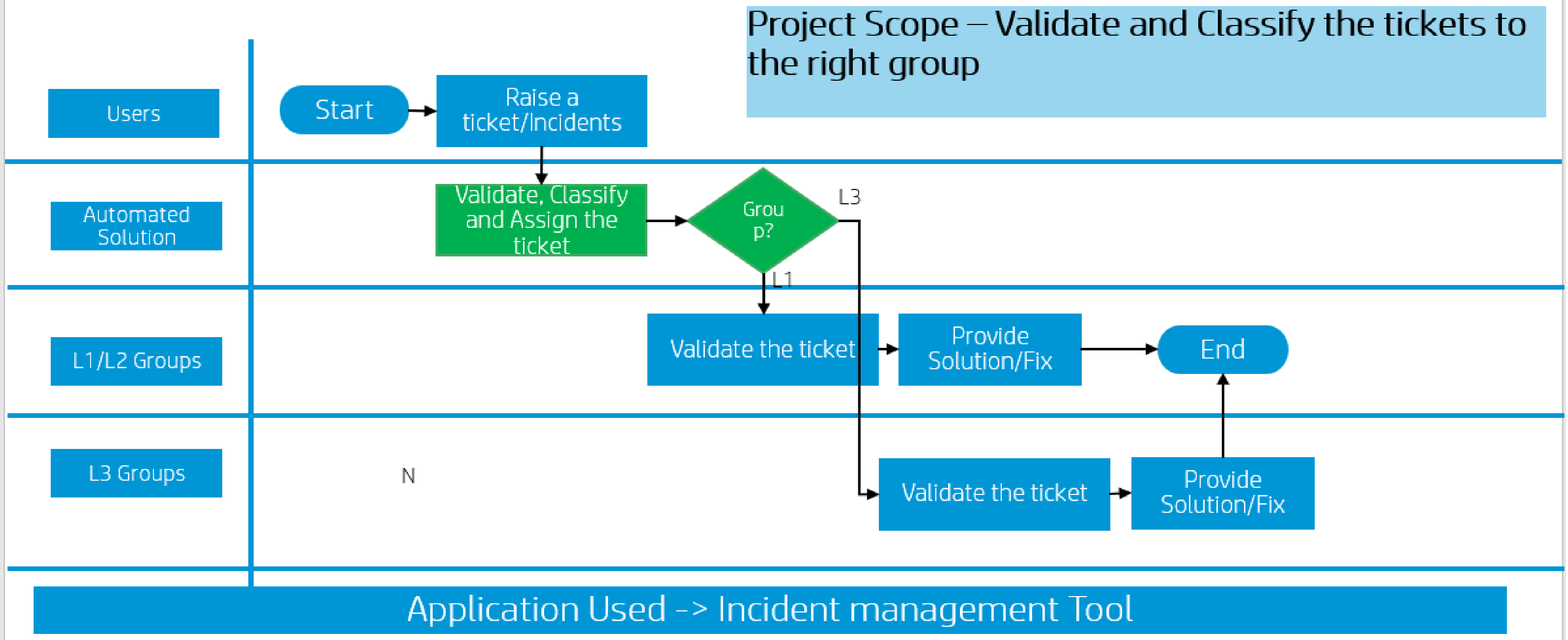
As-is Process Map

Capstone Project : Automatic Ticket Assignment – AS IS



To-be Process Map

Capstone Project : Automatic Ticket Assignment – TO BE(Ideal Scenario, Assuming we have a very good accuracy in the automated solution)

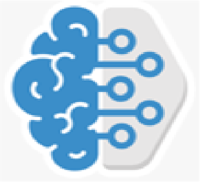


Approach To Be Taken

Capstone Project : Model Building Approach/Steps

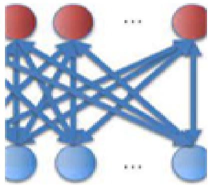
Pre-Processing, Data Visualization and EDA

- Exploring the given Data files
- Understanding the structure of data
- Finding inconsistencies in the data
- Visualizing different patterns
- Visualizing different text features
- Text pre-processing /Data Clean-up
  - Dealing with missing values
  - Removal of duplicate requests
  - Stop word removal
  - Removal of punctuations, numbers, special characters
  - Tokenization
  - Combining Desc, Short Desc
  - Lemmatization
  - Summary



Model Building

- Building model based on classification ML algorithms
  - Naïve Bayes
  - KNN
  - Bagging Classifier
  - Support vector machine
  - Decision Tree
  - Random Forest
- Deep Learning Models
  - LSTM
- Accuracy comparison and summary



Test the Model Fine Tuning and Repeat

- Different Model building with evaluation metrics – Milestone2
- Hyperparameter tuning
- Reporting evaluation metrics

Exploratory Data Analysis

Overview

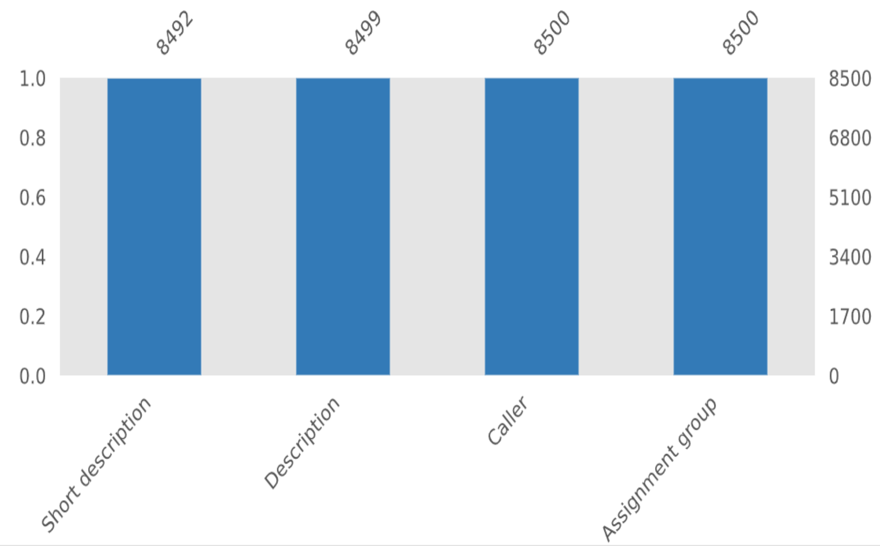
Dataset statistics

Number of variables	4
Number of observations	8500
Missing cells	9
Missing cells (%)	< 0.1%
Duplicate rows	83
Duplicate rows (%)	1.0%
Total size in memory	4.3 MiB
Average record size in memory	532.4 B

Highlights:

- Data contains 8500 rows and 4 columns
- There are total 9 missing values in the entire data set which is less than 0.1%. Hence can be ignored for this project purpose
- There are 83 duplicates rows in the dataset. We can analyse these duplicates further.

Missing Value



Highlights:

- There are total 9 missing values in the entire data set
- 8 missing values are noted in Column 'Short description'
- 1 missing value is observed in Column 'Description'

Action to be taken:

- As mentioned above, since dataset has less than 0.1% missing value. Missing values removed during pre-processing.

Exploratory Data Analysis

Duplicates

	Short description	Description	Caller	Assignment group
51	call for ecwtrjnq jpecxuty	call for ecwtrjnq jpecxuty	olckhmvx pcqobjnd	GRP_0
229	call for ecwtrjnq jpecxuty	call for ecwtrjnq jpecxuty	olckhmvx pcqobjnd	GRP_0
493	ticket update on inplant_872730	ticket update on inplant_872730	fumkcsji sarmtlhy	GRP_0
512	blank call //gso	blank call //gso	rbozivdq gmlhrtvp	GRP_0
667	job bkbackup_tool_powder_prod_full failed in j...	received from: monitoring_tool@company.com\r\n...	bpctwhsn kzqsbmtp	GRP_8
...	...	...	...	...
7836	probleme mit erpgui \tmqfjard qzhgdoua	probleme mit erpgui \tmqfjard qzhgdoua	tmqfjard qzhgdoua	GRP_24
8051	issue on pricing in distributor_tool	we have agreed price with many of the distribu...	hbmwlprq ilfvyodx	GRP_21
8093	reset passwords for prgthyuulla ramdntythanjes...	the	boirqctx bkijgqry	GRP_17
8347	blank call // loud noise	blank call // loud noise	rbozivdq gmlhrtvp	GRP_0
8405	unable to launch outlook	unable to launch outlook	wjtzrmqc ikqpbflg	GRP_0

83 rows × 4 columns

Highlights:

- Out of 8500 rows, there are 83 duplicates identified in the dataset; which is approx. 1% of the data.
- We are assuming that these duplicate tickets are created multiples times due to some glitch or may be for seeking for updated or prioritization.

Action to be taken:

- During pre-processing these duplicates rows have been dropped based on above assumption.

Exploratory Data Analysis

Reviewing features

Short description
Categorical

HIGH CARDINALITY

Distinct count	7481
Unique (%)	88.1%
Missing	8
Missing (%)	0.1%
Memory size	66.5 KiB

password reset	38
windows password reset	29
account locked in ad	24
windows account locked	23
erp SID_34 account unlock	19
Other values (7476)	8359

Description
Categorical

HIGH CARDINALITY

Distinct count	7817
Unique (%)	92.0%
Missing	1
Missing (%)	< 0.1%
Memory size	66.5 KiB

the	56
windows password reset	29
password reset	26
windows account locked	23
account locked in ad	23
Other values (7812)	8342

Caller
Categorical

HIGH CARDINALITY

Distinct count	2950
Unique (%)	34.7%
Missing	0
Missing (%)	0.0%
Memory size	66.5 KiB

bpctwhsn kzqsbmtp	810
ZkBogxib QsEJzdZO	151
fumkcsji sarmtlhy	134
rbozivdq gmlhrtvp	87
rkupnshb gsmzfojw	71
Other values (2945)	7247

Assignment group
Categorical

HIGH CARDINALITY

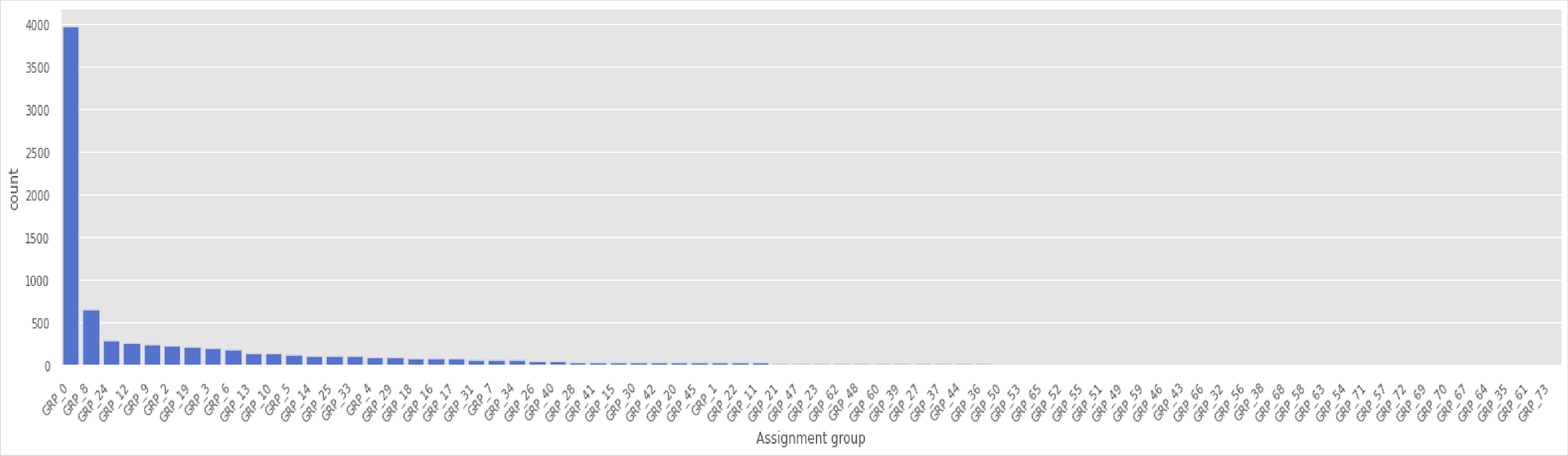
Distinct count	74
Unique (%)	0.9%
Missing	0
Missing (%)	0.0%
Memory size	66.5 KiB

GRP_0	3976
GRP_8	661
GRP_24	289
GRP_12	257
GRP_9	252
Other values (69)	3065

Exploratory Data Analysis

5 point Summary of Assignment group

count	74.000000
mean	114.864865
std	465.747516
min	1.000000
25%	5.250000
50%	26.000000
75%	84.000000
max	3976.000000



Highlights:

- There are 74 unique groups for whom the tickets are assigned in Assignment group column
- Group_0 has almost 50% of the records, there is a class imbalance
- There are top 3 callers who have raised most tickets - which can be looked at from a user experience matrices perspective
- Looking at the same issue assigned to various groups, it is good for multiclass classification
- There is no significant relation between callers and tickets assigned so we can remove the caller column
- From looking at the data manually, there are certain tickets and groups which follows a pattern and caller is always assigned to a specific group, having rules build based on regular expressions can take care of the issues.
- There are certain tickets which is non English - Has to be translated before building ML model

Pre – Processing

Steps taken for pre-processing:

1. Feature selection:

- Duplicates appearing in 83 rows were removed
- Column ‘Caller’ to be dropped
- Created new feature by combining text from ‘Short Description’ and ‘Description’

2. Data Cleaning and Processing:

✓ **Lower case -**

- Text was converted into lower cases

✓ **Creating function for data cleaning-**

- Created ‘`is_valid_date`’ function and ‘`clean_data`’ function to remove email addresses from the description columns as it will not add any value for classification, to remove numeric values which might be a problem while preparing model for classification, to remove all punctuations, special characters, double space was replaced by single space.

✓ **StopWord and Word Cloud**

- Used Stopword functionality to remove stop words to **eliminate words** that are so commonly used that they carry very little useful information
- Leveraged ‘WordCloud technique to further understand new column which was created by combining text from ‘Short Description’ and ‘Description’. WordCloud is a visualization technique for text data wherein each word is picturized with its importance in the context or its frequency. This is a very handy application when it comes to understanding the crux of text data.

Pre – Processing

Steps taken for pre-processing:

✓ **Tokenization:**

- Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.
- By applying tokenization on description column each words of sentence was converted into tokens.

Before Tokenization:

```
0     login issue verified user details employee man...          GRP_0
1     outlook hello team meetings skype meetings etc...          GRP_0
```

After Tokenization:

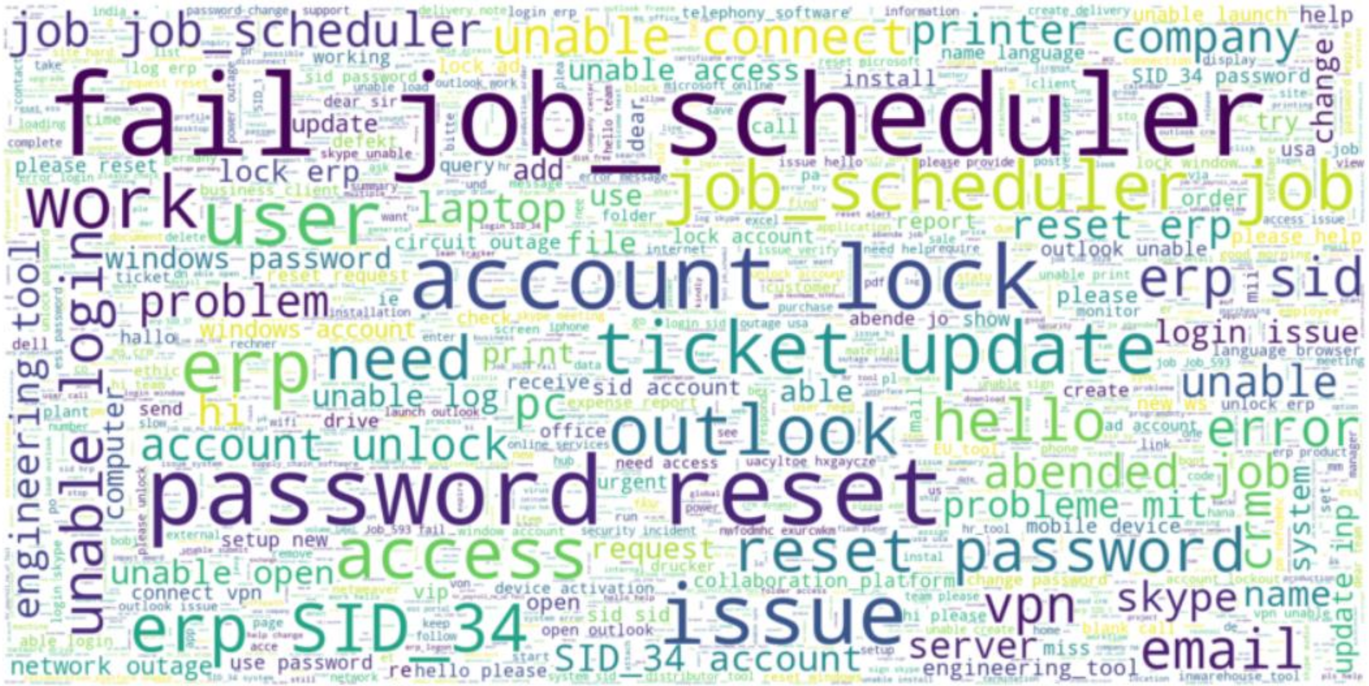
```
0        [login, issue, verified, user, details, employ...
1        [outlook, hello, team, meetings, skype, meetin...
```

✓ **Lemmatization:**

- Lemmatization is a method responsible for grouping different inflected forms of words into the root form, having the same meaning.
- For example – in above case, verified >>> verify

Pre – Processing

Final WordCloud post pre-processing of description:



Highlights:

- As demonstrated in WordCloud it is evident that maximum number of tickets are related to job scheduler, password, reset, login, email etc and are assigned to group_0
- Most of the tickets can be reduced by developing self help chat bot or ivr or scripts to resolve the issues
- After combining the short description & description fields a more comprehensive vocabulary is generated. But this caused sentences to have foreign language & English being combined. This became a deterrent in language translation.
- On checking model accuracy we observed negligible difference when 'Short Description' was included.
- Similar while performing spell check also, it was observed that model accuracy was not improving. Hence decided against doing this considering the huge time taken to perform the task by pyspellchecker.

Pre – Processing

Steps taken for pre-processing:

✓ N-gram model:

- An N-gram language model predicts the probability of a given N-gram within any sequence of words in the language.

Uni-gram

Summary	
job	3487
password	2291
erp	2077
fail	1746
user	1663
company	1585
reset	1520
issue	1518
unable	1496
access	1452

Bi-gram

Summary	
job scheduler	949
fail job	789
password reset	786
fail job_scheduler	785
00 job	761
job job	617
job fail	472
account lock	459
backup circuit	441
reset password	354

Tri-gram

Summary	
fail job scheduler	789
job fail job	461
job job fail	460
00 job job	450
fail job_scheduler 09	308
fail job_scheduler 10	285
group acl inside	185
src inside dst	185
access group acl	185
fail job_scheduler 08	171

Highlights:

- Both WordCloud and N-gram model are similar words with highest frequency, such as –fail job scheduler, password reset and others

Pre – Processing

Final dataframe creation:

✓ **Step 1:**

Based on the manual analysis of the text,few groups can be assigned directly to where model is not required, so we are droppi ng the rows which contains the below groups,and will consider only balance groups for model building for better accuracy on t icket assignment

GRP_35 need access to erp need access to
GRP_38 delete the charm project fy_13
GRP_39 space available memotech space consumed
GRP_43 shop_floor_app production order number
GRP_46 erp nx9
GRP_51 product selector credit component monitoring_tool
GRP_54 logical warehouse reduce stock level
GRP_55 finance_app how to run the report from finance_app
GRP_57 failed in job_scheduler i was able to access this before
GRP_58 job qeue job processor
GRP_61 internal error: unable to find any processes calibration system
GRP_64 change in report not an otc report only used by your finance team
GRP_66 installing cutview update cutview
GRP_67 complete forecast complete my forecast
GRP_68 expense report not working expense report will not submit
GRP_69 repeat outbound connection for 135/tcp expense report will not submit
GRP_70 repeat outbound connection for 135/tcp create signature
GRP_71 na production files not received not received the production feed files
GRP_72 update to anftgup nftgyair account locked
GRP_73 sso portal on the hub oneteam sso not working

✓ **Step 2:** Groups with less than 100 were grouped.

Final 18 Groups

	GRP_0	GRP_A	GRP_8	GRP_24	GRP_12	GRP_9	GRP_2	GRP_19	GRP_3	GRP_6	GRP_13	GRP_10	GRP_5	GRP_14	GRP_25	GRP_33
Group_ID	3934	1376	645	285	257	252	241	215	200	183	145	140	128	118	116	107

Model Building

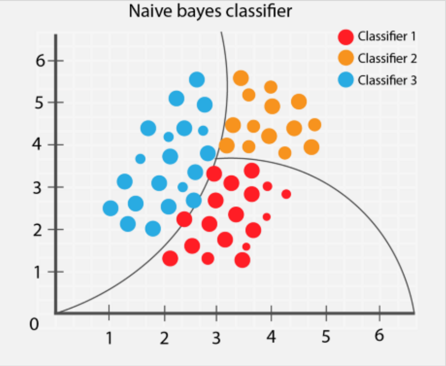
1. Final data was divided into train and test data with 80:20 equation
2. Classification function was created to run multiple classification models
3. Below are the models which will be built and validated for the best

   - **Multinomial Naive Bayes -** Naive Bayes is based on Bayes' theorem, where the adjective Naïve says that features in the dataset are mutually independent. Occurrence of one feature does not affect the probability of occurrence of the other feature. For small sample sizes, Naïve Bayes can outperform the most powerful alternatives. Being relatively robust, easy to implement, fast, and accurate, it is used in many different fields.
     - **In this dataset,** features are independent and dataset is small in size. Hence Naïve Bayes can be a good option to explore here.

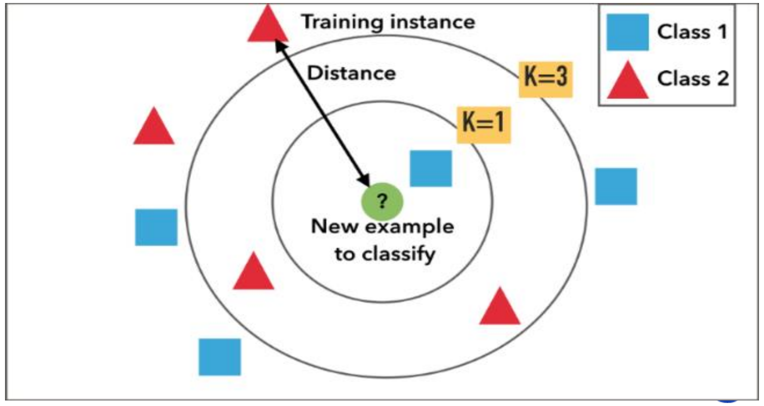
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

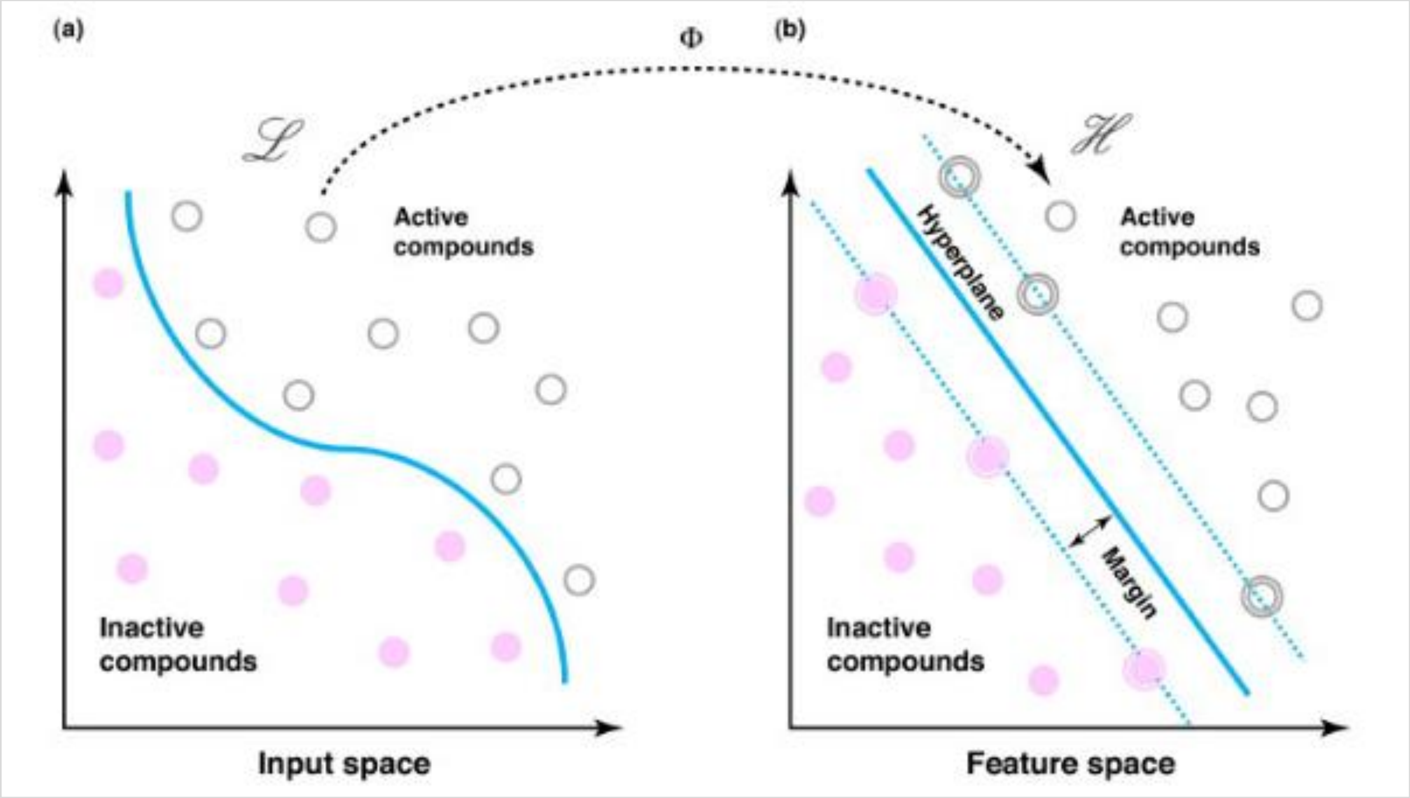


   - **K Nearest neighbor (KNN) -** KNN algorithm is used to classify by finding the K nearest matches in training data and then using the label of closest matches to predict. Mostly Euclidean distance is used to find the closest match. Generally, neighbors share similar characteristics and behavior that's why they can be treated as they belong to the same group.
     - **In this text dataset,** similar characteristics and behavior of neighbors can be a looked into for understanding the issue for which ticket was raised. Hence KNN can be a good option to explore here.



Model Building

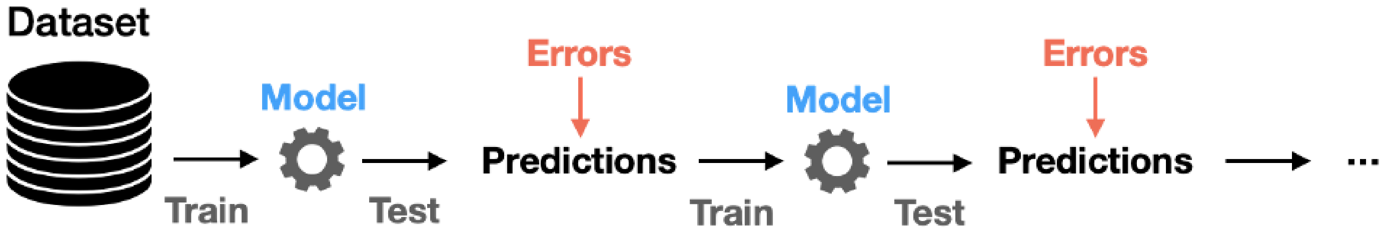
- **Support Vector Machine -** The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features.
  - **In this dataset,** maximum number of tickets are related to job scheduler, password, reset, login, email etc and are assigned to one group. However, we have seen similar issue assigned to different group. SVM works well in linearly separable dataset but its worth checking performance of SVC on this dataset as SVM applies hinge loss. Also, SVC only concerns for small instance to the border or discriminator and avoids the other examples



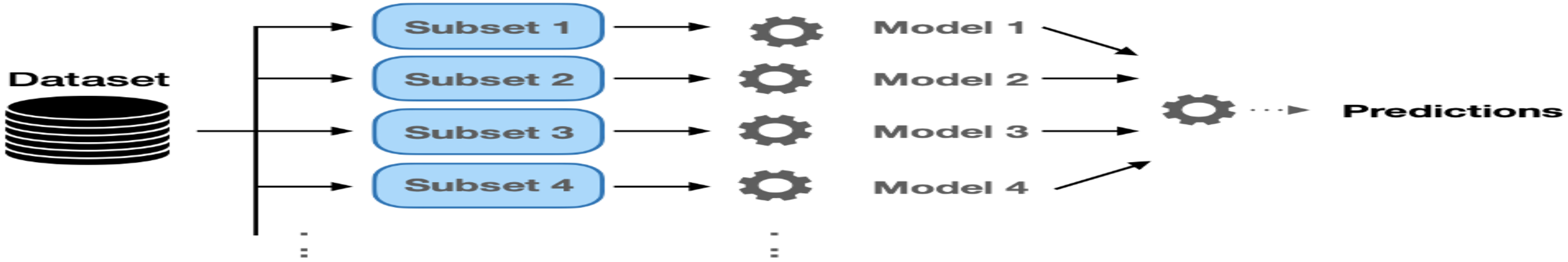
Model Building

Ensemble Techniques - Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. It is a multistep learning to categorize and predict the regularized classes and novel classes utilizes the feature evolution extraction and concept evolution extraction methods. **In this dataset**, its worth checking performance of different Ensemble techniques as combination of multiple models can produce improved results.

- **Decision Tree** - A decision tree classifies inputs by segmenting the input space into regions.
- **Random Forest** - The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification. An RF model comprises a set of decision trees each of which is trained using random subsets of features. The prediction by the RF is obtained via majority voting of the predictions of all the trees in the forest.
- **AdaBoost Classifier –** It is an ensemble of algorithms, where we build models on the top of several weak learners. Sequential decision trees were the core of such adaptability where each tree is adjusting its weights based on prior knowledge of accuracies.



- **Bagging Classifier -** Bagging is based on a bootstrapping sampling technique. Bootstrapping creates multiple sets of the original training data with replacement. Replacement enables the duplication of sample instances in a set. Each subset has the same equal size and can be used to train models in parallel.



Great Learning

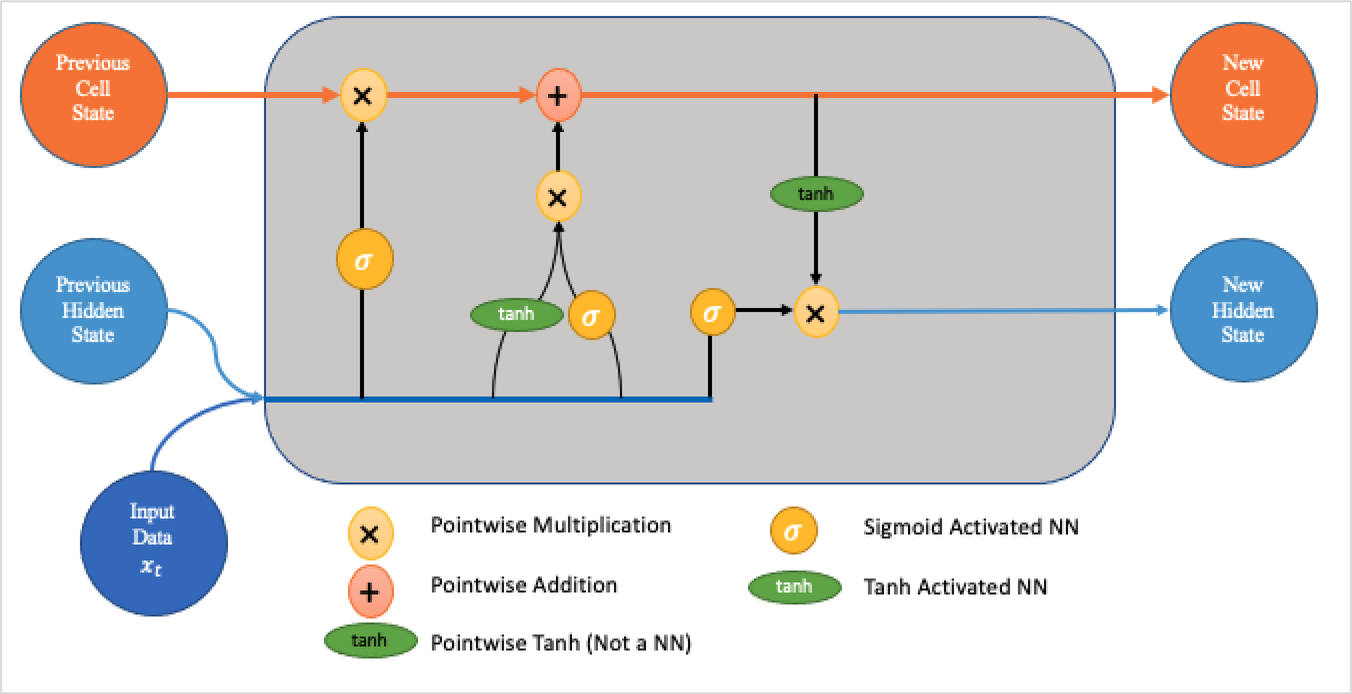
Model Building

LSTM –

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

LSTMs use a series of 'gates' which control how the information in a sequence of data comes into, is stored in and leaves the network. There are three gates in a typical LSTM; forget gate, input gate and output gate. These gates can be thought of as filters and are each their own neural network.

Since **the dataset** is small in size and hence leveraging the capability of Neural Network can be a good option.



Great Learning

Model Building

Model Performance:

✓ **Below are the accuracy results for the built models**

Multinomial Naive Bayes - Training - 62.51%, Testing - 58.66%

K Nearest neighbor (KNN) - Training - 76.67%, Testing - 67.65%

Support Vector Machine - Training - 96.33%, Testing - 73.22%

Decision Tree - Training - 99.73%, Testing - 65=3.15%

Random Forest - Training - 99.73%, Testing - 68.36%

AdaBoost Classifier - Training - 51.27%, Testing - 49.91%

Bagging Classifier - Training - 99.70%, Testing - 68.72%

LSTM - Training - 89.21%, Testing - 62.55%

Way Forward:

- As per the above accuracy from the classifier models, we see that data is overfitting due be below reasons even though the data was cleaned to a certain extent 1) Data is highly imbalance due to skewness in the data which is for Group_0 2) Data contains non English words (Assumption)
- In Milestone2 we will be dealing with Imbalanced data and fine tuning the models to make if right fit and classifies the tickets to the right group with certain degree of accuracy