Automatic Ticket Assignment

CAPSTONE PROJECT GROUP - B3

Nilesh Honavar, Mohit Saran, Venkatesh Venkatakrishnan & Venumadhav Anandaramu

Table of Contents

SUMMARY OF PROBLEM STATEMENT, DATA AND FINDINGS

BUSINESS CONTEXT:

Business Domain: Incident Management process in an IT function

Purpose: The main goal of Incident Management process is to provide a quick fix / workarounds or solutions that resolves interruption and restores service to its full capacity to ensure no business impact

Source: Incidents are created by various stakeholders (Business Users, IT Users and Monitoring Tools) within IT Service Management (ITSM) Tool and assigned to Service Desk teams (L1 / L2 teams)

As – Is Process Workflow:

Incidents from ITSM Tool are assigned to Service Desk teams (L1 / L2 teams)

This team will review incidents for right ticket categorization, resolve or re-assign to other Functional teams from Applications and Infrastructure (L3 teams) as needed

L1 / L2 needs to spend time (~ 15mins per incident) reviewing Standard Operating Procedures (SOPs) before assigning to Functional teams (Minimum ~25-30% of incidents needs to be reviewed for SOPs before ticket assignment)

There are 74 functional groups in which above tickets get classified into

As – Is Process Shortcomings:

Assignment of incidents to appropriate IT functional groups is manual process

74 functional groups seem to be a very high number in any Ticket management setup and may also be driving process inefficiencies leading to incorrect allocations

Manual assignment of incidents is inefficient and not cost effective

Human errors during ticket assessment based on description leading to Incidents being assigned to incorrect functional group

Increases response & resolution times resulting in user satisfaction deterioration / poor customer service

Problem Statement Observations:

Given dataset lacks important information provided by ITSM tools such as Incident Severity and SLA Priority which are primarily used for assigning tickets to different functional groups

Absence of above information prompts us to assume that given Group Structure of 74 groups is accurate and below hypothesis need to be proved to permit use of 'Assignment Group' as Target column for classification purposes:

OBJECTIVE:

Build an automated classifier that can classify tickets into appropriate functional groups by analyzing ticket description and augment/replace manual ticket categorization process, which will help organizations to reduce issue resolution time thereby can focus on more productive tasks and substantial cost savings.

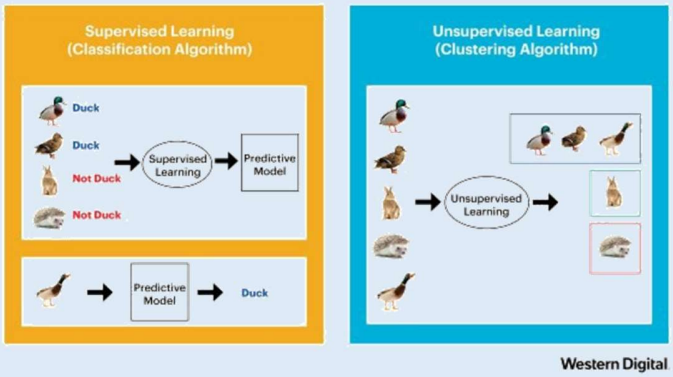
Goal #1: Uses hidden pattern in ticket description to auto-identify appropriate functional group

Goal #2: Verify that classifier works as designed for unseen tickets during test

SOLUTION:

Text in ticket description can be an extremely rich source of information for classification, but extracting insights from it can be hard and time-consuming due to its unstructured nature

Apply Natural language processing (NLP) and other AI-guided techniques can be used to classify text in a faster, more cost-effective, and more accurate manner. Based on pros and cons of Text classification techniques below



- MNB and SVN will be suitable for Supervised Learning on this dataset
- Topic Modelling will be suitable for Non Supervised Learning on this dataset

Types of NLP tasks for text classification and underlying assumptions:

1. Supervised machine learning - Most popular machine learning algorithms for creating text classification models include Multinomial Naive Bayes (MNB), support vector machines (SVM), and deep learning

- Conditions to apply Supervised Learning:
  - Pre-defined label (also called target) must have labels correctly defined and minimal class imbalance
  - MNB - One can get really good results even when your dataset isn't very large (~ a couple of thousand tagged samples) and computational resources are scarce
  - SVM doesn't need much training data to start providing accurate results. SVM algorithms are "multi-dimensional." So, the more complex the data, the more accurate the results will be. SVM does, however, require more computational resources than Naive Bayes, but results are even faster and more accurate
  - Deep learning (RNN and LSTM) architectures offer huge benefits for text classification because they perform at super high accuracy with lower-level engineering and computation. Deep learning algorithms do require much more training data than traditional machine learning algorithms (at least millions of tagged examples). However, they don't have a threshold for learning from training data, like traditional machine learning algorithms, such as SVM and Deep learning classifiers continue to get better the more data you feed them with.

2. Unsupervised machine learning techniques (Text Modelling, K-means clustering) can be used when a dataset to be analyzed does not have labels or existing label structure requires re-classification

- Conditions to apply Non-Supervised Learning:
  - Number of documents plays most important role; it is theoretically impossible to guarantee identification of topics from a small number of documents
  - Length of documents plays a crucial role: poor performance of the Topic Models like LDA is expected when documents are too short, even if there is a very large number of them. Ideally, documents need to be sufficiently long, but need not be too long topics
  - In theory, the convergence rate deteriorates quickly to a nonparametric rate, depending on number of topics used to fit the LDA. This implies, the user needs to exercise extra caution to avoid selecting overly large number of topics for the model. Similarly, K means algorithm is not capable of determining optimum number of clusters. Finding the optimum number of clusters is a challenge
  - LDA performs well when underlying topics are well-separated in sense of Euclidean metric. Another favorable scenario is concerned with distribution of documents within topic polytope: when individual documents are associated mostly with small subsets of topics, so that they are geometrically concentrated

OVERVIEW OF THE FINAL PROCESS: EXECUTION STRATEGY

- State hypothesis to check if given Labels are valid and can be considered for supervised learning
- Explore given dataset to understand data distribution patterns, data inconsistencies and text features influencing incident distribution across groups
- Analyze data distribution patterns, data inconsistencies and feature engineering requirements
- Text Preprocessing to clean and prepare data for feature extraction
- To Approve or Reject Hypothesis and determine if Target column can be considered for text classification task using Supervised or Unsupervised Learning
    - Apply Supervised Learning on Target labels of original dataset
        - Try different modelling techniques and capture results from these experiments
        - Verify if Test Accuracy and F1 score is above 75% (considering L1/L2 teams reallocate 25% incidents to other functional teams post review)
        - Approve Hypothesis if above is True, identify optimal model and tune hyper parameters and Deploy model for test data
- If Hypothesis Rejected – Apply Unsupervised Learning (Topic Modelling) to obtain new Target Labels
    - Try different modelling techniques and capture results from these experiments
    - Identify optimal model and tune hyper parameters
    - Verify Topic Classification as per new labels using Supervised Learning technique
    - Deploy model for test data

HYPOTHESIS:

Consider Supervised Learning for Labels in Assignment Group only if below is true

- Each group has distinct type of incidents. There is minimal/no overlap of incidents across groups
- Incident distribution across groups (aka classes) should not have significant imbalance

EXPLORATORY DATA ANALYSIS (EDA)

8500 rows and 4 columns

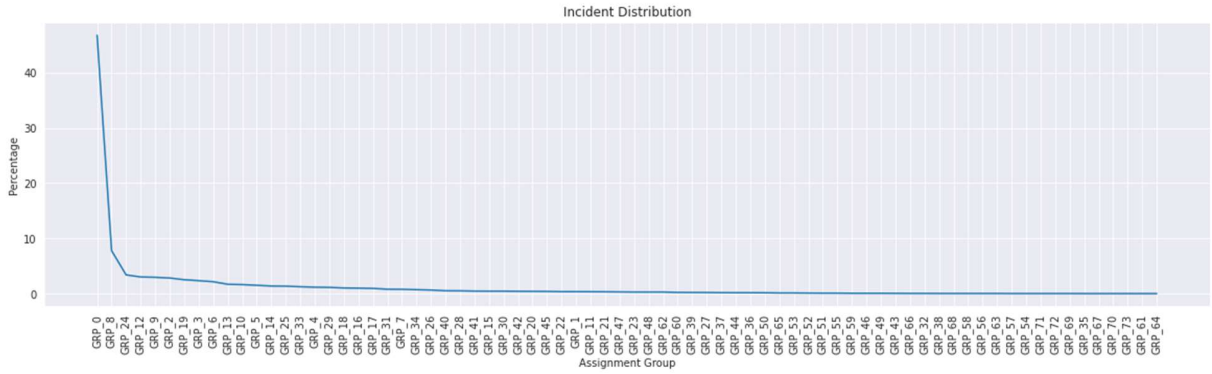
- There are four columns in dataset - Short Description, Description, Caller and Assignment group
  Short description and Description: Both these fields have mix of human input (emails, messages, etc.) and system tool output (alerts, notifications, etc.) and highlight the unstructured text
    - There is a presence of null values which will be replaced by NAN
    - Visual inspection shows non-English language, symbols, spaces, junk characters, presence of usernames, Email ID, greetings, system acronyms and application names. These will be treated using appropriate Data cleansing techniques listed in the sections below
    - There are spelling mistakes and typos found due to manual input of ticket description which need to be taken care
    - Duplicates affect model accuracy and will need to be treated
    - For few tickets short Description and Description are same, also we observed short description repeated along with Description We will be taking care of this as well
- Caller: ID of the user reporting incident is not relevant for text classification purposes and will be dropped
- Assignment Group: Given Group labels for the incident.
    - 74 Assignment Groups – Seems to be a lot of groups for the number of tickets provided in the dataset
    - Group labels will need to be analyzed to determine if it is suitable for classification based on Hypothesis approval/rejection

Dataset descriptors -

	Short description	Description	Caller	Assignment group
count	8492	8499	8500	8500
unique	7481	7817	2950	74
top	password reset	the	bpctwhsn kzqsbmtp	GRP_0
freq	38	56	810	3976

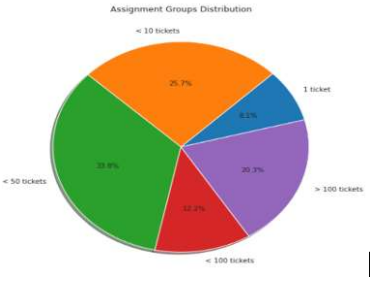
'GROUP' - LABEL DISTRIBUTION

1. Incident distribution across Groups is highly imbalanced - GRP_0 @ ~47% , GRP_8 @ 8%, Majority groups averaging below 3% with some of them under 1%
    a. **IMPLICATION** – Imbalanced classifications pose a challenge for predictive modeling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class
    b. Additional steps would be needed to treat imbalance through feature engineering – merging, resampling



2. Distribution of the tickets across groups
    - 25 Groups range between 1 - 10 tickets |
    - 25 Groups range between in 10 - 50 tickets
    - 9 Groups range between 50 - 100 tickets
    - 15 Group has 100+ tickets

	Description	Ticket Count
0	1 ticket	6
1	< 10 tickets	19
2	< 50 tickets	25
3	< 100 tickets	9
4	> 100 tickets	15

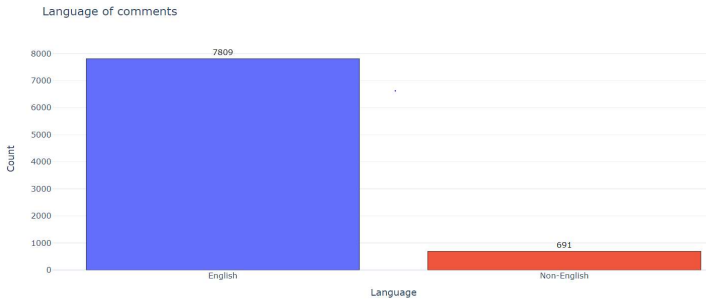


• TREATMENT OF TEXT INCONSISTENCIES IN SHORT DESCRIPTION AND DESCRIPTION

	INCONSISTENCY	IMPLICATION	IMPUTATION
1	Null values	Affects accuracy adversely	There were 8 Null values in Short description and 1 Null value in Description, which was replaced by empty string

2	Presence of caller names within text description	Verbose and does not add meaningful features to the vocab. Can Impact pre-set embedding. Also impacts translation of non English	`Number of rows in short description having caller name:381`
			`Number of rows in description having caller name:1839`
			Replaced by empty string
			`Count of Caller names in Short_description column : 0`
			`Count of Caller names in Description column : 0`

3	Presence of unrecognized unicode characters	Encoder Algorithms cannot be applied and will affect accuracy	FTFY package used to convert into good Unicode helping identify other Non-English language like Mandarin and rectifying German umlauts.

			SD_no_Caller windows ç³»ç»Ÿæ– æ³•ç™»å¼•æ¤ç¤ºè®¡ç®—æœºä¸ŽåŸŸä¹‹é—´ä¿¡ä»»å…³ç³»å¤±è´¥ Desc_no_Caller windows ç³»ç»Ÿæ– æ³•ç™»å¼•æ¤ç¤ºè®¡ç®—æœºä¸ŽåŸŸä¹‹é—´ä¿¡ä»»å…³ç³»å¤±è´¥ SD_clean windows 系统无法登录提示计算机与主域之前信任关系失败 Desc_clean windows 系统无法登录提示计算机与主域之前信任关系失败 SD_no_Caller support fÃ¼r we111\ Desc_no_Caller support fÃ¼r we111\ SD_clean support für we111\ Desc_clean support für we111\
4	Detecting Non English languages	Most data pre-processing techniques like STOP WORDS, LEMMATIZATION are best used for English Language	Using pyglot package, detected 691 rows which had Non-english languages in description fields  array(['English', 'Latin', 'Kinyarwanda', 'Malay', 'Chinese', 'German', 'Scots', 'Dutch', 'Welsh', 'Norwegian Nynorsk', 'Polish', 'Danish', 'un', 'Portuguese', 'Japanese', 'Malagasy', 'Icelandic', 'Tagalog', 'Nauru', 'Luxembourgish', 'Spanish', 'Turkish', 'Waray', 'Finnish', 'Interlingua', 'Hungarian', 'French', 'Norwegian', 'Latvian'], dtype=object)

TEXT PRE-PROCESSING STEPS:

All steps listed below are to be followed in order during Training -

	Steps	Technique	IMPLICATION	Remarks
1	Translation of Non-English Languages to English	Explored Google Translate API	Majority of non-English words have been converted into English language SD_clean windows 系统无法登录提示计算机与主域之前信任关系 Desc_clean windows 系统无法登录提示计算机与主域之前信任关系 SD_clean_translate windows system can not login prompt before the... Desc_clean_translate windows system can not login prompt before the...	There is limit imposed from Google on number of requests for any unpaid service, so this cannot be used. This can be used in deployment using paid service
		Used Google Sheets		Offline Language translation cannot be included in the deployment pipeline
2	Data Cleansing	Regular Expressions and removal of manually identified verbose	Removing presence of unnecessary verbose, continuous phrases, greetings, email body, email IDs, tags, dates and numbers. Excludes irrelevant features from Vocabulary and also avoids curse of dimensionality as well as aids in text classification SD_clean_translate unable to connect to companysecure at usa, oh SD_clean unable to connect to companysecure at usa oh Name: 7387, dtype: object Desc_clean_translate contact # \nno one at the site is able to co Desc_clean contact no one at the site is able to connec	This has helped to detect more duplicates indicated by drop in unique values below count unique SD_clean_translate 8500 7262 SD_clean 8500 6702 Desc_clean_translate 8500 7694 Desc_clean 8500 6971
3	Spell Checking	SYMSPELL package with custom Tech Word Dictionary	Spelling correction helps in cleaning text to assist in removing duplicate words and Lemmatization process improving Vocabulary quality. Tech Acronyms/ Application names were added to custom dictionary to avoid unwanted alteration SD_clean_translate distributor_tool not laodir SD_clean distributor tool not loadir Name: 8434, dtype: object	It takes significant time for correction of spelling mistakes in data and hence may not be included during deployment This has helped to detect further duplicates indicated by drop in unique values below count unique SD_clean_translate 8500 7262 SD_clean 8500 6018 Desc_clean_translate 8500 7694 Desc_clean 8500 6340
4	Removing duplicates from short description and description columns	Spotted common text in Description columns	Reduces dataset size to 6613 rows from earlier 8500 from removal of common row duplicates. Number of common rows for Short description and description fields : 188	This step follows Spell Checking as prior helped rectify several spelling errors within text description allowing for them to be detected as duplicates and removed
5.	Feature engineering of Columns	Merging Short and Description columns into new column 'Clean Description' and Dropping 'Caller' column	2179 rows have identical Short Description and Description fields - When merging, only Short Description is considered to avoid verbose SD_clean unable to log in to engineering tool and sky Desc_clean unable to log in to engineering tool and sky Clean_Description unable log engineering tool sky Name: 5, dtype: object 4434 rows have distinct values in Short and Description fields – When merging, they will be merged in the sequence of 'Desc + Short Desc' SD_clean why we can use the outlook grou Desc_clean be Clean_Description best why we can use the outlook grou Name: 4070, dtype: object	Merging of columns has improved context and completeness wherever description in individual columns were inadequate count unique Clean_Description 6613 6613 SD_clean 6613 6018 Desc_clean 6613 6340 Duplicate rows also impact word embeddings when using BOW or TF-IDF algorithms and hence need to be dropped
6	STOP WORDS	Extended NLTK STOP Words set	It was observed that words used for greetings, salutations, etc should be added to STOP words set as they were irrelevant features in Vocabulary leading to curse of dimensionality and could have adversely impacted classification	101 STOP words removed is an addition to earlier removal of words / phrases/ sentences has detected duplicates count unique Clean_Description 6613 6511

<table>
<tr>
<td>7</td>
<td>Removal of repeat words within merged Description field</td>
<td>Custom function to drop repeat words</td>
<td>4193 repeat words get removed .improving word count statistics
which may help in BOW and TF IDF algorithms
SD_clean outlook
Desc_clean meetings skype meetings not appearing outlook ...
Clean_Description meetings skype meetings not appearing outlook ...
CD_norepeatwords meetings skype not appearing outlook calendar ...
Name: 1, dtype: object
<table>
<tr><th></th><th>Nonrepeat</th><th>Withrepeat</th></tr>
<tr><td>count</td><td>6613.000000</td><td>6613.000000</td></tr>
<tr><td>mean</td><td>13.117042</td><td>18.871314</td></tr>
<tr><td>std</td><td>17.237011</td><td>35.705387</td></tr>
<tr><td>min</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>25%</td><td>5.000000</td><td>5.000000</td></tr>
<tr><td>50%</td><td>9.000000</td><td>11.000000</td></tr>
<tr><td>75%</td><td>16.000000</td><td>21.000000</td></tr>
<tr><td>max</td><td>432.000000</td><td>751.000000</td></tr>
</table></td>
<td>We observed that removing duplicate words distorts contextual placement of words which adversely affects Topic Modelling algorithm accuracy.
Due to these concerns, duplicate words will not be removed</td>
</tr>
<tr>
<td>8</td>
<td>STEMMING</td>
<td>Porter stemmer</td>
<td>Stemming is causing last character of multiple words to be omitted which is not correct in the context of the domain as it may distort acronyms and technical words
<table>
<tr><th></th><th>CD_norepeatwords</th><th>CD_St</th></tr>
<tr><td>0</td><td>verified user details employee name checked ad...</td><td>verifi user detail employe name check ad rese</td></tr>
<tr><td>1</td><td>meetings skype not appearing outlook calendar ...</td><td>meet skype not appear outlook calendar somebo</td></tr>
<tr><td>2</td><td>cannot log vpn</td><td>cannot log v</td></tr>
<tr><td>3</td><td>unable access hr tool page</td><td>unabl access hr tool pa</td></tr>
<tr><td>4</td><td>skype error</td><td>skype er</td></tr>
<tr><td>...</td><td>...</td><td></td></tr>
<tr><td>8495</td><td>not receiving remails sent z mail advise coming</td><td>not receiv remail sent z mail advis co</td></tr>
<tr><td>8496</td><td>telephony software issue</td><td>telephoni softwar is</td></tr>
<tr><td>8497</td><td>vip windows password reset</td><td>vip window password re</td></tr>
<tr><td>8498</td><td>unable access machine utilities finish drawers...</td><td>unabl access machin util finish drawer adjust</td></tr>
<tr><td>8499</td><td>various not opened multiple pics conc area</td><td>various not open multipl pic conc ar</td></tr>
</table></td>
<td>While Stemming helped increase detect duplicates,
<table>
<tr><th></th><th>count</th><th>unique</th></tr>
<tr><td>Clean_Description</td><td>6613</td><td>6511</td></tr>
<tr><td>CD_Stem</td><td>6613</td><td>6476</td></tr>
</table>
Due to distortion concerns, Stemming
will not be used for this problem statement</td>
</tr>
<tr>
<td>9</td>
<td>LEMMATIZATION</td>
<td>WordNET Lemmatizer with POS Tagging</td>
<td>Lemmatization has improved removal of similar meaning words getting them to their root words which reduces Vocab size to addresses 'curse of Dimensionality'
Clean_Description hostname currently experiencing high cpu utilization investig
CD_Lem_token_combined hostname currently experience high cpu utilization investig
Name: 474, dtype: object

Post Lemmatization, number of records that have reduced their word length are: 2</td>
<td>Wordnet created tokens have resulted in false detection of 110 rows having Non-English words.
On closer inspection, we will keep them as-is with no further need of data transformation</td>
</tr>
<tr>
<td>10</td>
<td>Removing Duplicates after Lemmatization</td>
<td>Dropping duplicates</td>
<td>130 rows have been removed post removal of duplicates
count 6613
unique 6483</td>
<td>After preprocessing, dataset ready for modelling has 6483 rows</td>
</tr>
</table>

DESCRIPTION - VOCABULARY SIZE & SENTENCE LENGTH

Above Text Pre-processing steps have helped in improving Vocab quality. Let us now look at the resultant dataset

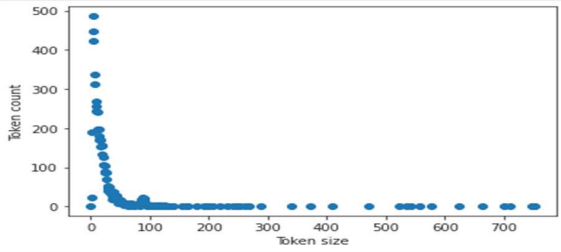
Revised Dataset Shape: 6474 rows (removed 1 row which had zero token value post lemmatization)

Vocab Size: Pre-Lemmatization Vocabulary Size: 7035 | Post Lemmatization Vocabulary Size: 5698

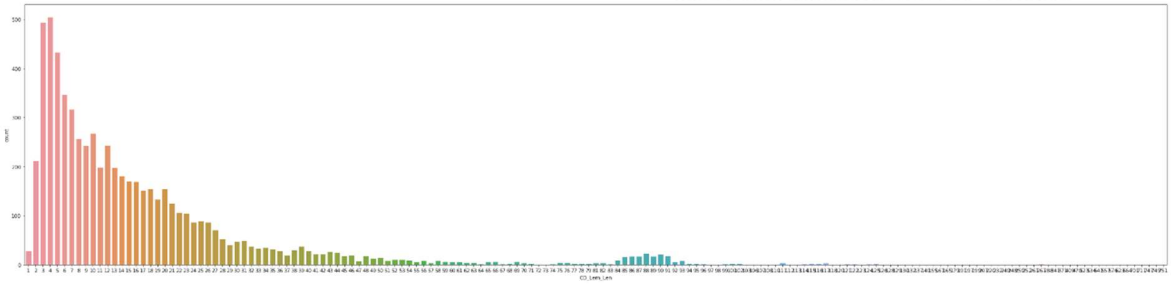
Sentence Length:

- Average 19.21 words are present in each sentence | 75% of records have less than 21 words per sentence – This will need to be factored in during selection of algorithm
- There are 24 rows have 1 word in the description field. These may be insufficient for ticket classification
- There are very few records that are in excess of 200 tokens with a max of 751 words

OBSERVATIONS: Since in an IT domain, most incidents occur in pairs (e.g.: password reset, account lock, job fail, etc), we shall explore if bigrams helps improve Vocabulary quality which may also result in increase of Vocab size




```
count    6474.000000
mean       19.193389
std        35.883676
min         1.000000
25%         5.000000
50%        11.000000
75%        21.000000
max       751.000000
Name: tokencount, dtype: float64
```



FACTORS CONSIDERED FOR EVALUATING HYPOTHESIS

1) Using Wordcloud to check if Groups have distinct incidents or incidents are highly interspersed

Word Cloud of Major groups have below incident details

GRP_0 - Most tickets related to User access, outlook issues, password reset, access issues, login issue, connectivity issue related to various tools

GRP_8 - More tickets related to outage, job failures, monitoring tool, telecom

GRP_24 - More tickets related to IT infra setups like - workstation, Printer ,skype,lan

GRP_12 - More tickets related to network related issues citrix, acls

GRP_9 has more tickets related to reports, analytics *and* job scheduler

Remaining Groups – Has a distribution of similar incidents observed in the major groups across multiple groups

Based on Wordclouds, there is a significant overlap of incidents appearing across multiple groups, For example reset password, erp, job scheduler even in other groups apart from top 5 groups

IMPLICATION: Each group has subset of multiple incident types which creates challenges for Text classification algorithms which are dependent on separation of classes content during Supervised Learning

1. Reviewing a single Incident type 'Job Scheduler' to understand distribution pattern

1. 981 instances of Job scheduler cross 14 different sub incident are spread across 23 different groups
2. There is no clear pattern emerging that an incident type is isolated to a particular Group

Analysis of a single Incident type ' Job scheduler' also reveals random spread across multiple groups

Count of Assignment group	Column Labels														
Row Labels	abended in job	apo	bk	bkbackup	bkwin	bwhrattr failed in job	HostName	hr	Job	mm	pp	SID	snp	Misc	Grand Total
GRP_8	54	10	16	24			16	4	202	10	29	46	4	2	417
GRP_9	31					8			135					10	184
GRP_6	31	5						1	70				14	3	124
GRP_5	16	7	6	4			2	1	21			53		2	112
GRP_10	2							36	20					4	62
GRP_60							7		4					8	19
GRP_12									8					5	13
GRP_47	8								1			1			10
GRP_45									7						7
GRP_1	1											5		1	7
GRP_14	2						1		1					2	6
GRP_13									4						4
GRP_29	2								1						3
GRP_18									3						3
GRP_2														2	2
GRP_65							1								1
GRP_44									1						1
GRP_3														1	1
GRP_51														1	1
GRP_57									1						1
GRP_20														1	1
GRP_4														1	1
GRP_40														1	1
Grand Total	147	5	17	22	28	8	27	42	479	10	29	105	18	44	981

2. Analyzing Using bigrams to identify clusters of words distributed amongst groups

- Number of bigrams generated – 11,20,680

> A bigram is a sequence of two adjacent elements from a string of tokens and imply the frequency of occurrence of these words occurring together is higher. For similar Incident types, one can assume that the words used to describe them should occur more frequently together and are unique Groups to aid in supervised learning process for text classification.
>
> Below analysis shows interspersion of bigrams across multiple groups

```
1  new_df[new_df['CD_Lem_token_combined'].str.contains("login issue")].Group.value_counts()
```

```
GRP_0     57
GRP_7      1
GRP_23     1
GRP_27     1
GRP_22     1
GRP_40     1
Name: Group, dtype: int64
```

```
1  new_df[new_df['CD_Lem_token_combined'].str.contains("reset password")].Group.value_counts()
```

```
GRP_0     141
GRP_17     31
GRP_2      16
GRP_7      10
GRP_53      3
GRP_21      1
GRP_12      1
GRP_31      1
GRP_19      1
Name: Group, dtype: int64
```

EVALUATING HYPOTHESIS – APPROVE OR REJECT?

APPLY SUPERVISED LEARNING ON TARGET LABELS OF GIVEN DATASET

- Popular Machine Learning and Deep Learning Algorithms used widely for Text Classification:
  1. **Stochastic Gradient Descent (SGD)** - Implements logistic regression and is expected to work well for text classification because it can deal with sparse data
  2. **Multinomial Naive Bayes (MNB) -** Expected to get good results even when dataset isn't very large (~ a couple of thousand tagged samples) and computational resources are scarce
  3. **Support Vector Machines (SVC)** - Powerful text classification machine learning algorithm, and doesn't need much training data to start providing accurate results. SVM does, however, require more computational resources than Naive Bayes, but the results are even faster and more accurate.
  4. **Random Forest (RF) classifiers** - Are suitable for dealing with 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
  5. **Bagging classifier** – Algorithm classifies results using number of base classifiers results voting combination classification model which has a better classification performance
  6. **XGBoost** - especially widespread because it has been the winning algorithm in a number of recent Kaggle competitions

7. **LSTM NN** – The intuition behind LSTM is that the machine will learn the importance of previous words, so that we will not lose information from the older hidden states. This makes them better at finding and exposing long range dependencies in data which is imperative for sentence structures.
8. **LSTM Bi-Directional** – Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems which is really putting two RNN's together. The structure allows the networks to have both backward and forward information about the sequence at every time step.
9. **GRU Model** - GRUs are improved version of standard recurrent neural network. To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output. They can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.
10. **COHEN KAPPA BENCHMARKING:** Cohen suggested the Kappa result be interpreted as follows: values ≤ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement

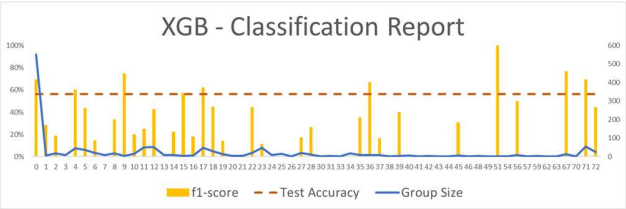
- Conducted Multi class Text Classification experiments with below Machine Learning Algorithms

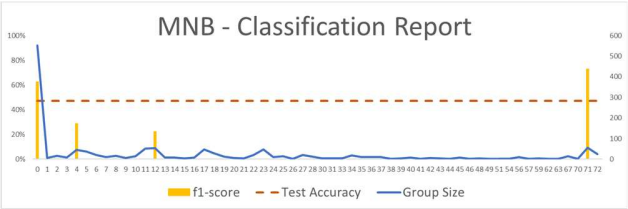
OBSERVATIONS:

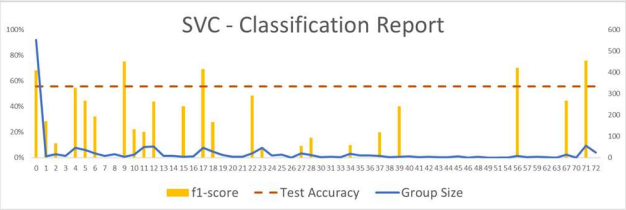
1. Significant difference between Training and Test Accuracy scores for SGD, RFC and KNN indicating that there is significant Overfitting and these algorithms may not be suitable for our purpose
2. MNB, RBF_SVC and XGB have more comparable Train and Test Accuracy scores

Model	Train_Acc	Test_Acc	F1_Score	Cohen_Kappa	CV_F1
SGD_Clf	0.89	0.57	0.56	0.47	0.57
MNB_Clf	0.52	0.50	0.65	0.13	0.36
RFC_Clf	1.00	0.59	0.67	0.35	0.50
linear_SVC	0.86	0.53	0.50	0.44	0.56
RBF_SVC	0.74	0.48	0.46	0.38	0.50
KNN	1.00	0.60	0.66	0.40	0.53
XGB_Clf	0.74	0.58	0.65	0.37	0.53

3. Inspecting Classification Report multi class F1 scores and determine if hyper tuning can improve results



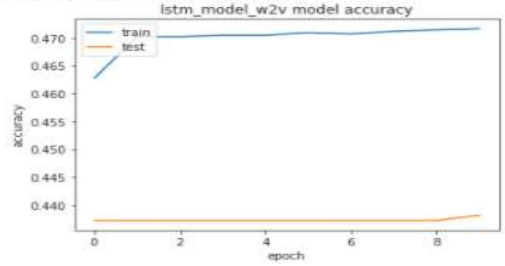




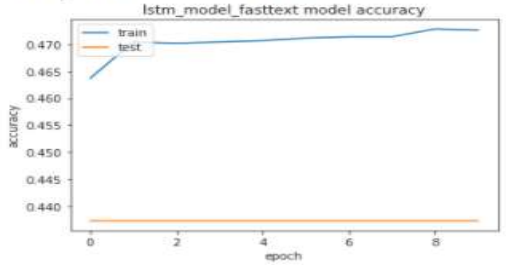

```
Cohen kappa score:
 0.3250450323173101
---------------------------
Training accuracy: 81.193%
---------------------------
Testing accuracy: 57.838%
---------------------------
```
Without hyperparameters in SGD

```
Cohen kappa score:
 0.46240134136072863
---------------------------
Training accuracy: 89.380%
---------------------------
Testing accuracy: 56.371%
---------------------------
```
With hyperparameters in SGD

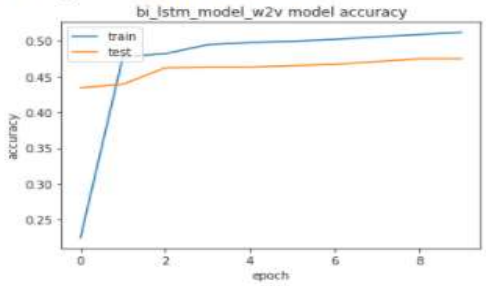
```
predicting accuracy of unidirectional LSTM model with word2vec
Accuracy is: 0.4555984555984556
```



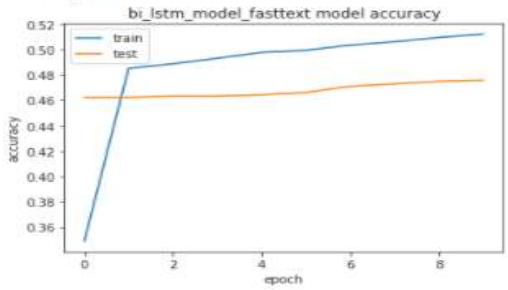

```
predicting accuracy of unidirectional LSTM model with fasttext
Accuracy is: 0.4548262548262548
```



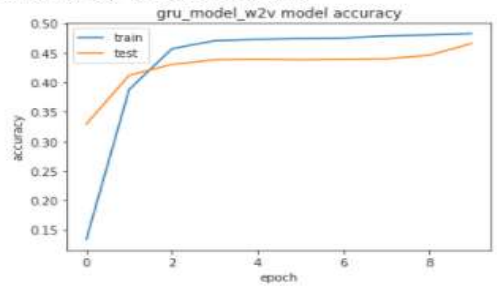
```
predicting accuracy of bidirectional LSTM model with word2vec
Accuracy is: 0.4911196911196911
```



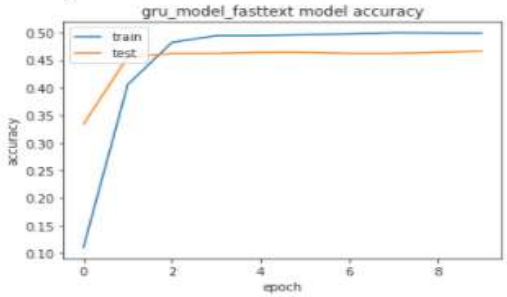

```
predicting accuracy of bidirectional LSTM model with fasttext
Accuracy is: 0.49266409266409267
```




```
predicting accuracy of bidirectional GRU model with word2vec
Accuracy is: 0.4749034749034749
```




```
predicting accuracy of bidirectional GRU model with fasttext
Accuracy is: 0.47876447876447875
```



INFERENCE:

- Test Accuracy scores across all experiments are < 60%
- Cohen Kappa Benchmark score for SGD Classifier was around 32% which is poor and after applying hyperparameters it was still 46% which is a moderate score.
- LSTM and GRU with its variants don't provide good score, So we are going with ML which are giving better scores.
- Heavy class imbalance resulting in classes having zero F1 scores limiting improvement opportunity post hyper tuning

HYPOTHESIS ASSESSMENT:

STATED HYPOTHESIS:

Consider Supervised Learning for Labels in Assignment Group only if below is true

- Each group has distinct type of incidents. There is minimal/no overlap of incidents across groups
- Incident distribution across groups (aka classes) should not have significant imbalance

ASSESSMENT FACTOR	HYPOTHESIS INVALID/VALID ?
Significant Imbalance observed for GROUP field	INVALID – Groups are heavily imbalanced. Group_0 has a huge chunk of tickets alone.
Using Word Cloud to check for spread of incidents within Groups	INVALID – Incidents of similar nature are present across several groups.
Reviewing a single Incident type 'Job Scheduler' to understand distribution pattern	INVALID – Job Schedular example issue is present in almost 50%> of tickets.
Analyzing Using bigrams to identify clusters of words distributed amongst groups	INVALID – Bigrams were also identified and were not found to be unique across groups.
Inferences from Supervised Model – Training and Test scores	INVALID – Poor scores of accuracies on these models (lesser than 75% classification done manually).

Above Hypothesis is REJECTED, based on significant class imbalance and inferences drawn from Word Cloud, Incidents spread across groups and Supervised Model Training and Test scores

We cannot consider given 'GROUP' field as Target column to apply Supervised Learning algorithms

We shall explore Unsupervised Learning (TOPIC MODELLING) techniques to derive appropriate Group clusters.

Once new Group clusters are derived, for further verification, we shall use Supervised Learning (TEXT CLASSIFICATION) to check allocation of tickets into these groups

DECIDING APPROACH FOR UNSUPERVISED LEARNING BY WEIGHING IN PROS/CONS

Pros outweigh the Cons for Scenario 1 and **'Topic Modelling'** will be used to reclassify incidents int new groups

	Scenario	Pros	Cons
1	Ignore current group structure of 74 groups and reclassify (Topic modelling) tickets based on common incident types across all groups. Observe new group structure with reduced number of groups, with good distribution of tickets across groups and reduced overlap of incident types	New group structure will have less imbalance of the incidents across groups. Lower class imbalance improves classification accuracy. Reduced overlap of incidents across groups. Tangible Business Benefit: Based on resource pool, appropriate groups with lower incident resolution SLAs can be further segregated into L1/L2 team. Those with higher priority/SLAs for incident resolution can be directly marked to L3 teams	Overrides existing group structure which may be based on current organizational factors not known to us. May need skill specific resource pool to handle unique incident type across classes - This may increase resource pool size
2	Inspect for most common issues across each Group that are less frequent in other groups Based on domain expertise, merge Groups that have similar incident types to reduce class imbalance until we reach 'X' number of groups with similar volume distribution If there are groups that continue to have lesser than 50 tickets, merge them into a single group New group structure will have reduced class count (X classes), however each class will have a combination of incident types that may be uniformly distributed	May reduce class imbalance due to merging of groups May reduce resource pool requirement as number of Groups will be lower than scenario 1 since multiple incident types will be clubbed in one group	Retains existing group structure which is observed to be inaccurate with multiple incidents interspersed across groups. Since each class will have a combination of incidents, Resources mapped to these classes will need to be multi skilled to handle varied incident/language type Since groups are getting merged based on volume and incident type, there is a high likelihood of similar incident types being distributed across several groups leading to inefficiency New incident type not trained for previously may be ignored

TOPIC MODELLING

Topic modelling is an unsupervised machine learning technique that automatically analyses text data to determine cluster words for a set of documents. It refers to the process of dividing a corpus of documents in two:

1. A list of the topics covered by the documents in the corpus
2. Several sets of documents from the corpus grouped by the topics they cover

Underlying assumption is that every document comprises a statistical mixture of topics, i.e. a statistical distribution of topics that can be obtained by "adding up" all of the distributions for all the topics covered. What topic modelling methods do is try to figure out which topics are present in the documents of the corpus and how strong that presence is.

Length of documents plays a crucial role: poor model output is expected when documents are too short, even if there is a very large number of them. Ideally, the documents need to be sufficiently long, but need not be too long topics

Types of Topic Modelling?

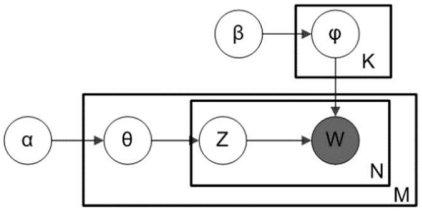
- **K Means clustering** - k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centre or cluster centroid), serving as a prototype of the cluster
    - **Pertinence**: Relatively simple to implement. Can warm-start the positions of centroids, Easily adapts to new examples, Generalizes to clusters of different shapes and sizes, such as elliptical clusters

- **Challenges**: Choosing optimal k value manually, Clustering data of varying sizes and density, Clustering outliers, Scaling with number of dimensions.

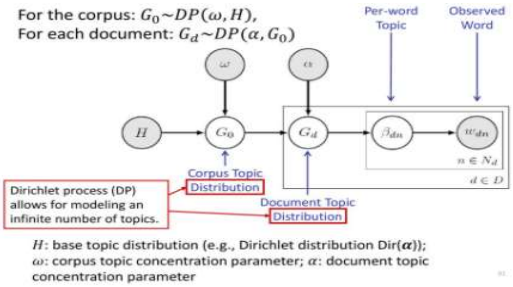
- **Latent Semantic Analysis(LSA)** a.k.a Latent Semantic Indexing - LSI examines the words used in a document and looks for their relationships with other words. LSI allows a system to determine the kind of words that a document might be relevant for, even if they are not actually used on the document itself. But having content that is full of words that have relationships with each other, you are strengthening the document for all of those words
  - **Pertinence:** Easy to implement, understand and use. LSA can handle Synonymy problems to some extent (depends on dataset though), Since it only involves decomposing term document matrix, it is faster, compared to other dimensionality reduction models
  - **Challenges**: It is a linear model, so not the best solution to handle non linear dependencies, The latent topic dimension cannot be chosen to arbitrary numbers. It depends on the rank of the matrix, so can't go beyond that, The model is not humanly readable. And not easy to interpret like LDA, Deciding on number of topics is based on heuristics and needs some expertise

- Based on desk research, K means clustering and Latent Semantic Analysis are not popularly used for Topic Modelling and hence due to time constraint, we will focus our attention to more popularly used Topic Modelling algorithms below.

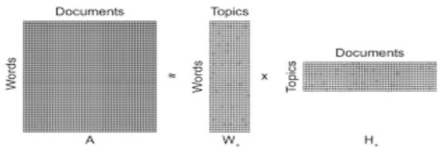
- **Latent Dirichlet Allocation (LDA)** - LDA is a significant extension of LSI. Words are grouped into topics. LDA assumes that each document in corpus contains mix of topics that are found throughout entire corpus. The topic structure is hidden - we can only observe the documents and words, not the topics themselves. Because the structure is hidden (also known as latent), this method seeks to infer the topic structure given the known words and documents. α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution, θm is the topic distribution for document m, φk is the word distribution for topic k, zmn is the topic for the n-th word in document m, and wmn is the specific word.
  - **Pertinence:** Most popular modelling technique used in the industry. Model is generative at document level and word level. The basic idea of this model is documents are represented as random mixtures over latent topics, where each topic is a distribution over words. LDA performs well when underlying topics are well-separated in sense of Euclidean metric. Does not suffer from overfitting issues like in LSI, Nosie reduction is possible by dimension reduction
  - **Challenges:** Need to choose number of Topics to be generated from the corpus. It is theoretically impossible to guarantee identification of topics from a small number of documents, Poor performance of LDA is expected when documents are too short, In theory, the convergence rate deteriorates quickly to a nonparametric rate, depending on number of topics used to fit the LDA. This implies, the user needs to exercise extra caution to avoid selecting overly large number of topics for the model.



- **Hierarchical Dirichlet Process (HDP)** - HDP is an extension of LDA, designed to address the case where the number of mixture components (the number of "topics" in document-modelling terms) is not known apriori. For HDP (applied to document modelling), one also uses a Dirichlet process to capture the uncertainty in the number of topics. So a common base distribution is selected which represents the countably-infinite set of possible topics for the corpus, and then the finite distribution of topics for each document is sampled from this base distribution
  - **Pertinence:** Maximum number of topics can be unbounded and learned from the data rather than specified in advance
  - **Challenges:** it is more complicated to implement, and unnecessary in the case where a bounded number of topics is acceptable.



- **Non-negative Matrix factorization** Non-Negative Matrix Factorization (NMF) is an unsupervised technique so there are no labelling of topics that the model will be trained on. The way it works is that, NMF decomposes (or factorizes) high-dimensional vectors into a lower-dimensional representation. These lower-dimensional vectors are non-negative which also means their coefficients are non-negative. Using the original matrix (A), NMF will give you two matrices (W and H). W is the topics it found and H is the coefficients (weights) for those topics. In other words, A is articles by words (original), H is articles by topics and W is topics by words. In Non-negative Matrix Factorization, a document-term matrix is approximately factorized into term-feature and feature-document matrices



  - **Pertinence:** One of the most popular and widely used technique. NMF is a deterministic algorithm which arrives at a single representation of the corpus. For this reason, NMF is often characterized as a machine learning algorithm. Both LDA and HDP are probabilistic models capable of expressing uncertainty about the placement of topics across texts and the assignment of words to topics, Thus, in cases where we believe that the topic probabilities should remain fixed per document (oftentimes unlikely)—or in small data settings in which the additional variability coming from the hyperpriors is too much—NMF performs better
  - **Challenges:** It fixes values for the probability vectors of the multinomials, whereas LDA allows the topics and words themselves to vary. This might result in NMF qualitatively leading to worse mixtures.

Short Text Topic Modeling (STTM)

Despite its great results on medium or large sized texts (>50 words), typically mails and news articles are about this size range, **LDA poorly performs on short texts** like Tweets, Reddit posts or StackOverflow titles' questions. Looking at the short texts examples in this Figure on the right, it is evident that the assumption made in LDA that a document is a mixture of topics is not true anymore.

We will now assume that a short text is made from **only one topic**. The **Gibbs Sampling Dirichlet Mixture Model** (**GSDMM**) is an "altered" LDA algorithm, showing great results on STTM tasks, that makes the initial assumption: **1 topic ↔1 document**. The words within a document are generated using the same unique topic, and not from a mixture of topics as it was in the original LDA.

- GSDMM algorithm would have been a great fit for the given Tickets dataset which has Avg sentence length @ ~ 13 words and potentially should include a single topic based on problem statement
- Based on desk research, since GSDMM is currently not completely evolved and does not have enough supporting articles, we are not able to experiment with GSDMM for this project.

COMMONLY USED PRACTICES FOR EVLUATING TOPIC MODELLING OUTPUT

- **Eye Balling Models:** Top N words | Topics / Documents
- **Evaluation Metrics:** How surprised a model is of new data it has not seen before (Perplexity Score) | Capturing model semantic similarity (Topic Coherence) | Topics interpretability
- **Human Judgements:** Relevance of keywords within a topic

- Recent studies have shown that predictive likelihood (or equivalently, perplexity) and human judgment are often not correlated, and even sometimes slightly anti-correlated
- Optimizing for perplexity may not yield human interpretable topics and thus Topic Coherence will be the preferred Evaluation Metric to be used in addition to Human Judgement

WHAT IS TOPIC COHERENCE?

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference

A set of statements or facts is said to be coherent, if they support each other. Thus, a coherent fact set can be interpreted in a context that covers all or most of the facts. An example of a coherent fact set is "the game is a team sport", "the game is played with a ball", "the game demands great physical efforts"

Let's take quick look at different coherence measures, and how they are calculated:

1. **C_v measure** is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity
2. **C_p** is based on a sliding window, one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence
3. **C_uci** measure is based on a sliding window and the pointwise mutual information (PMI) of all word pairs of the given top words
4. **C_umass** is based on document cooccurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure
5. **C_npmi** is an enhanced version of the C_uci coherence using the normalized pointwise mutual information (NPMI)
6. **C_a** is based on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity

- Of the two major types of Coherence Scores (C_V and C_UMass), we will consider C_V topic coherence score measure
- C_V typically ranges between 0 < x < 1
- General interpretation of C_V is as follows:
    - 0.3 is bad | 0.4 is low | 0.55 is okay | 0.65 might be as good as it is going to get | 0.7 is nice | 0.8 is unlikely and .9 is probably wrong

STEP-BY-STEP WALK THROUGH OF THE SOLUTION

Some unsupervised models like K-means clustering and LSA were tried, however based on research articles, other advanced techniques (LDA,HDP and NMF) were found to be more suited for Topic Modelling .

Topic Models evaluated for our Problem Statement:

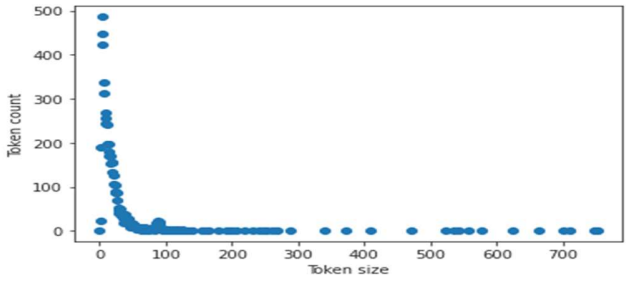
1. Latent Dirichlet Allocation (LDA)
    a. Gensim standard LDA which uses Variational Bayes sampling method
    b. Java based Mallet (Machine Learning for Language Toolkit) which uses Gibbs Sampling method
2. Hierarchical Dirichlet Process (HDP)
3. Non-Negative Matrix Factorization (NMF)

OUR EVALUATION CRITERIA FOR BENCHMARKING ONE OF THE TOPIC MODELS:

- **Coherence Score (c_v):** Score of 0.55 and above will be acceptable
- **Inspect topics by looking at highest-likelihood words in each topic:** Do they sound like they form a cohesive "topic" or just some random group of words
- **Inspect topic assignments quality:** Hold out a few random documents from training and see what topics model assigns to them. Manually inspect the documents and the top words in the assigned topics. Does it look like the topics really describe what the documents are actually talking about?

MODELLING STRATEGY USED:

- **Data Cleaning and Preprocessing** – Use output from Text cleaning and Preprocessing steps above. Data ready for model building has undergone treatment for null values, fixing bad Unicode characters, translation to English, Spell check, Data cleansing using Regular expressions, STOP words, removal of duplicates, dropping repeat words, lemmatization




```
count    6482.000000
mean       19.219994
std        35.959543
min         1.000000
25%         5.000000
50%        11.000000
75%        21.000000
max       751.000000
Name: tokencount, dtype: float64
```

- **Choice of Corpus – Unigrams and Bigrams**
  - We will use both Unigrams and Bigrams to check optimum suitability
    - While standalone words like application names, System acronyms also hold up on their own to determine Topics, in IT domain, incidents also occur in pairs (e.g.: password reset, account lock, job fail, etc.),
    - With bigrams, additional 794 meaningful words get added to the vocabulary which can help in getting better topics

```
Total Post Lemmatization Vocabulary Size: 5688
Total Post Lemmatization and Bigrams Vocabulary Size: 6482
```

- **Preparing Document Term Matric for feature extraction from corpus using BoW and TF-IDF**
  - What is BoW and TF-IDF?
    - Bag of Words (BOW) creates a vocabulary of all unique words occurring in all documents
    - TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus

```python
#function to create dictionary and corpus using Bag Of Words

def create_dict_corpus_bow(data_words):
# Create Dictionary
  id2word = corpora.Dictionary(data_words)

# Create Corpus from post clean data
  texts = data_words
# Term Document Frequency and Bag of words
  corpus = [id2word.doc2bow(text) for text in texts]

  return id2word, texts, corpus
```

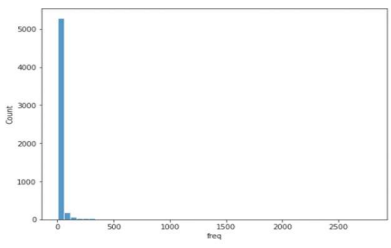
```python
#function to create dictionary and corpus using tfidf

def create_dict_corpus_tfidf(data_words):

  id2word, texts, corpus = create_dict_corpus_bow(data_words)

  tfidf = models.TfidfModel(corpus)
  corpus_tfidf = tfidf[corpus]

  return id2word, texts, corpus_tfidf
```

```python
# Create dictionary and corpus using single words and BoWs
id2word, texts, corpus = create_dict_corpus_bow(data_words)

# Create dictionary and corpus using bigram and BoWs
id2word_bigram, texts_bigram, corpus_bigram = create_dict_corpus_bow(data_words_bigrams)

# Create dictionary and corpus using single words and tfidf
id2word, texts, corpus_tfidf = create_dict_corpus_tfidf(data_words)

# Create dictionary and corpus using bigram and tfidf
id2word_bigram, texts_bigram, corpus_bigram_tfidf = create_dict_corpus_tfidf(data_words_bigrams)
```

  - Top 10 words appearing across documents based on frequency

Unigram

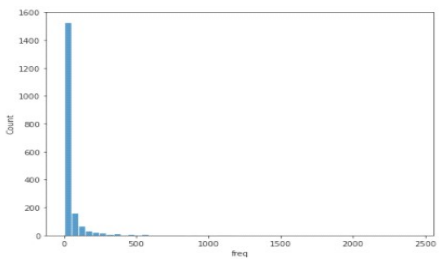
```python
dict_df = plot_word_freq(id2word=id2word, corpus=corpus)
```




```
no          2790
erp         1571
password    1472
user        1360
tool        1311
access      1308
issue       1231
work        1105
company     1088
error        986
Name: freq, dtype: int64
```

Bigrams

```python
dict_df_bigram = plot_word_freq(id2word=wordclouds_bigram.split(), corpus=corpus_bigram)
```



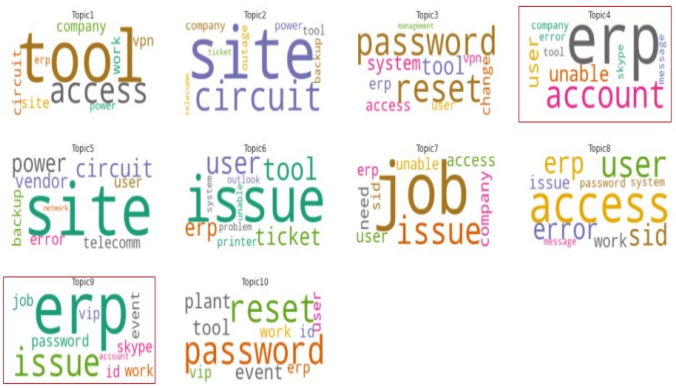
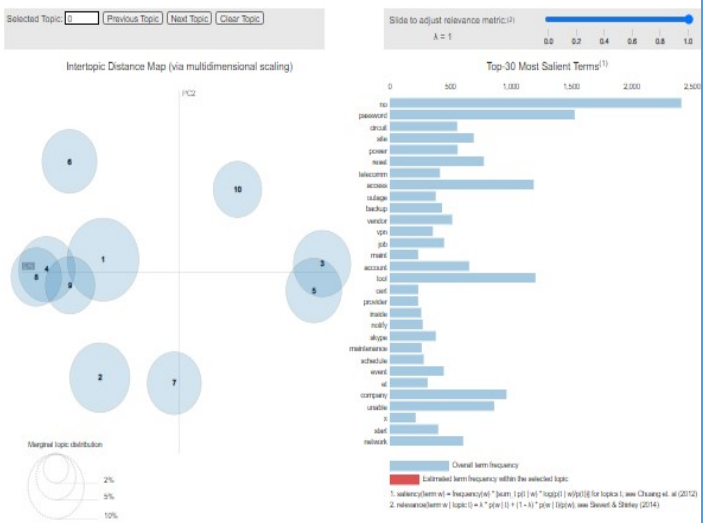
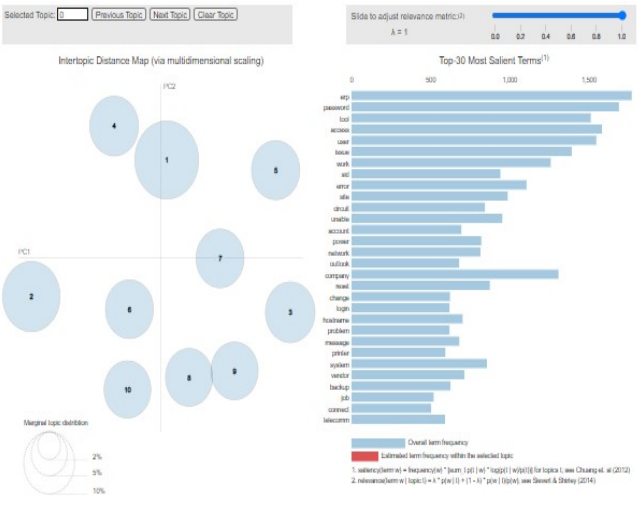

```
'hr_tool',      2434
'user',         2120
'closing',      1876
'no',           1502
'log',          1345
'check_caller', 1191
'advise',       1158
'use',          1124
'sid',          1119
'need',         1098
Name: freq, dtype: int64
```

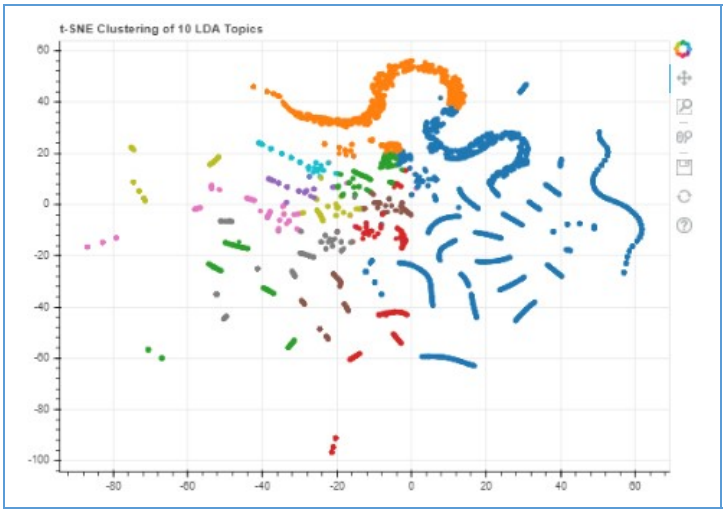
Baselining Model: LDA comparison for 10 Topics

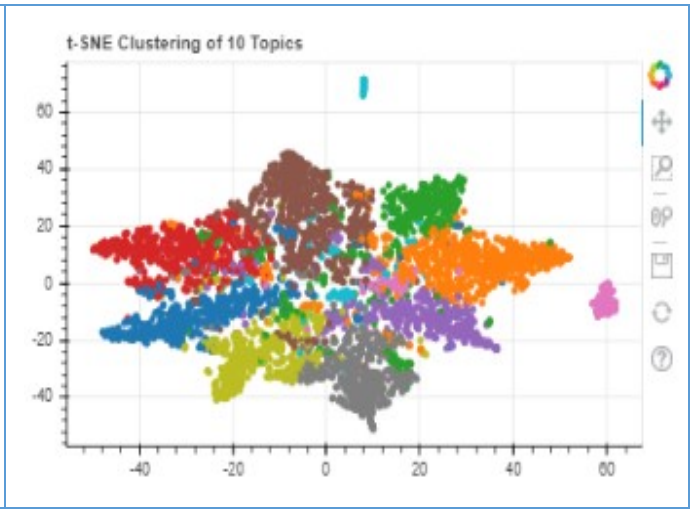
- Comparing Bayesian LDA and LDA Mallet output for unigram for BoW with respect to the Evaluating Criteria
  - Coherence Score (c_v) measures: **Mallet scores 0.5 vs. Bayesian LDA scores 0.44**
  - Inspect topics by looking at highest-likelihood words in each topic: **Topics generated by Mallet look more cohesive and demonstrate better separation of topics in comparison to Bayesian LDA**

BAYESIAN LDA (UNIGRAM & BAG OF WORDS)	LDA MALLET (UNIGRAM & BAG OF WORDS)
COHERENCE SCORE	

`num_topics = 10`
`model_Lda = LdaMulticore(corpus=corpus,`
`id2word=id2word,`
`num_topics=num_topics,`
`workers=4,`
`random_state=123,`
`iterations=1000,`
`per_word_topics=True)`
 	
`Num Topics = 10 LDA model has Coherence Value of 0.44`
	`num_topics = 10`
	`model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path,`
	`corpus=corpus,`
	`num_topics=num_topics,`
	`id2word=id2word,`
	`random_seed=123,`
	`iterations=100`
	`)`
	
	`Num Topics = 10 LDA Mallet has Coherence Value of 0.5`

INSPECT TOPICS BY LOOKING AT HIGHEST-LIKELIHOOD WORDS IN EACH TOPIC	
 **We see topics are spread and not distinct in comparison to lda mallet**	
	





OBSERVATION:

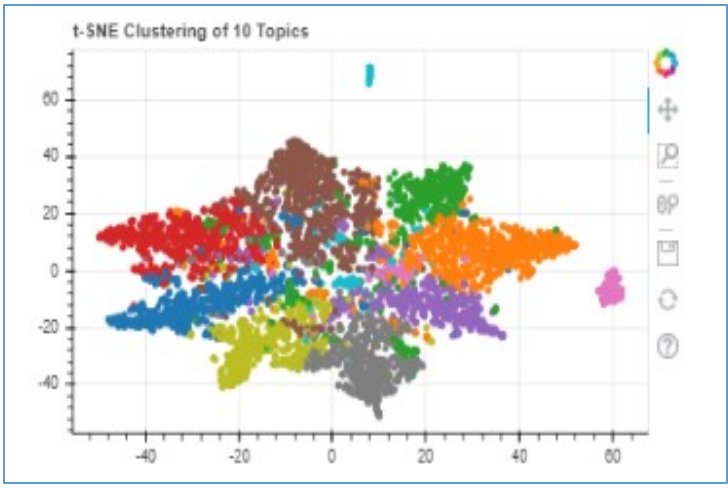
1. **Baseline LDA Mallet Unigrams + Bag of Words model** for comparison with other models (HDP and NMF) using the evaluation criteria
2. Model which emerges superior amongst the 3 will be picked to evaluate model version based on:
   a. Performance with Bigrams
   b. Performance with TF – IDF
3. The most optimal model version will be selected for tuning of Hyperparameters

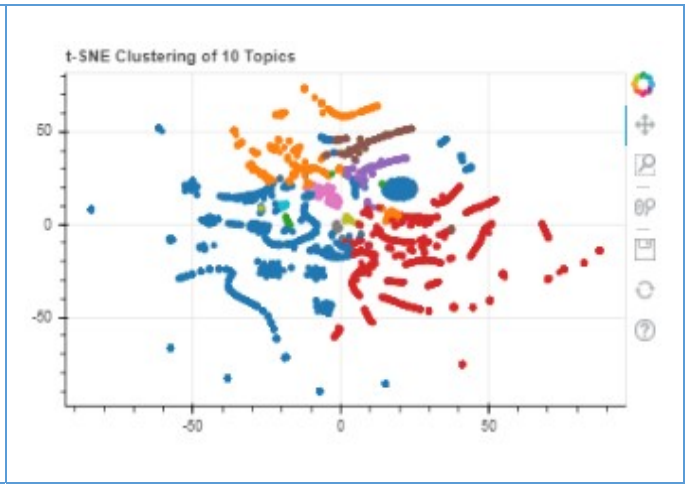
COMPARING: LDA Mallet (Unigrams + BoW) vs. HDP (Unigrams + BoW)

LDA MALLET (UNIGRAM & BAG OF WORDS)	HDP (UNIGRAM & BAG OF WORDS)
COHERENCE SCORE	

`num_topics = 10`
`model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topics=num_topics, id2word=id2word, random_seed=123, iterations=100 )`
`Num Topics = 10  LDA Mallet has Coherence Value of 0.5`
	`model_hdp = HdpModel(corpus=corpus, id2word=id2word, random_state=123)`
	`Num Topics = 10  HDP has Coherence Value of 0.66`

INSPECT TOPICS BY LOOKING AT HIGHEST-LIKELIHOOD WORDS IN EACH TOPIC	
	
	We see topics are spread and not distinct in comparison to lda mallet





OBSERVATIONS:

Comparing LDA Mallet with HDP output for unigram for BoW with respect to the Evaluating Criteria

a. Coherence Score (c_v) measures: LDA Mallet scores 0.50 vs. HDP scores 0.66

b. Inspect topics by looking at highest-likelihood words in each topic:

- Topics generated by Mallet look more cohesive and one can clearly identify the topics make more sense based on keywords within the topic.
- Some keywords like ['erp', 'tool', 'site'] seem to repeat frequently across several topics which will adversely impact text classification of such incidents
- T-SNE cluster plot shows better separation of topics for LDA Mallet in comparison to HDP

INFERENCE:

Based on above observations, LDA Mallet is more suitable for the given problem statement vs. HDP

We will proceed to compare LDA Mallet with NMF model

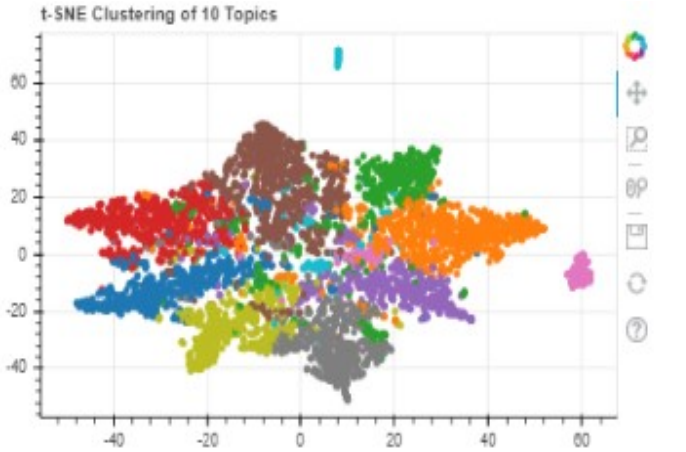
COMPARING: LDA Mallet (Unigrams + BoW) vs. NMF (Unigrams + BoW)

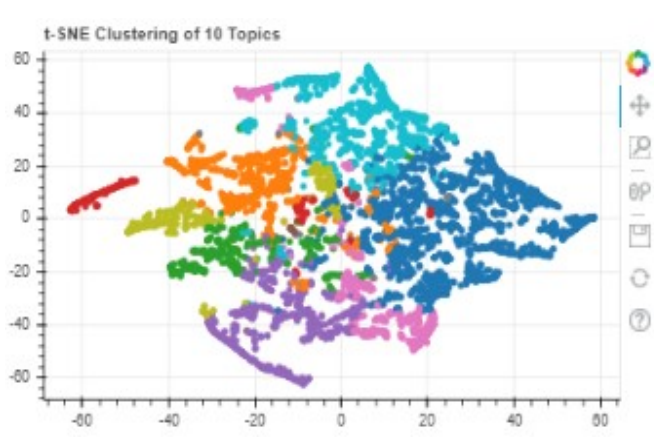
LDA MALLET (UNIGRAM & BAG OF WORDS)	NMF (UNIGRAM & BAG OF WORDS)
COHERENCE SCORE	

`num_topics = 10`
`model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path, corpus=corpus, num_topics=num_topics, id2word=id2word, random_seed=123, iterations=100 )`
`Num Topics = 10  LDA Mallet has Coherence Value of 0.5`
	`num_topics = 10`
	`model_nmf = Nmf(corpus=corpus, num_topics=num_topics, id2word=id2word, random_state=123)`
	`Num Topics = 10  NMF has Coherence Value of 0.65`

INSPECT TOPICS BY LOOKING AT HIGHEST-LIKELIHOOD WORDS IN EACH TOPIC	

We see topics are spread and not distinct in comparison to lda mallet





OBSERVATIONS:

Comparing LDA Mallet with NMF output for unigram for BoW with respect to the Evaluating Criteria

a. Coherence Score (c_v) measures: LDA Mallet scores 0.5 vs. NMF scores 0.65

b. Inspect topics by looking at highest-likelihood words in each topic:

- Both models seem to have produced similar keywords, however overall coherence of keywords to form a topic is more intuitive in LDA Mallet
- T-SNE cluster plot shows better separation of topics for LDA Mallet in comparison to NMF

INFERENCE:

Based on above observations, LDA Mallet is more suitable for the given problem statement vs. NMF

We will proceed to compare LDA Mallet variations in below order

1. Unigram with Bag of Words vs. Bigram with Bag of Words
2. Best of above Bag of Words vs. Unigram/Bigram TFIDF

COMPARING: LDA Mallet (Unigrams + BoW) vs. LDA Mallet (Bigrams + BoW)

LDA MALLET (UNIGRAM & BAG OF WORDS)	LDA MALLET (BIGRAMS & BAG OF WORDS)
COHERENCE SCORE	

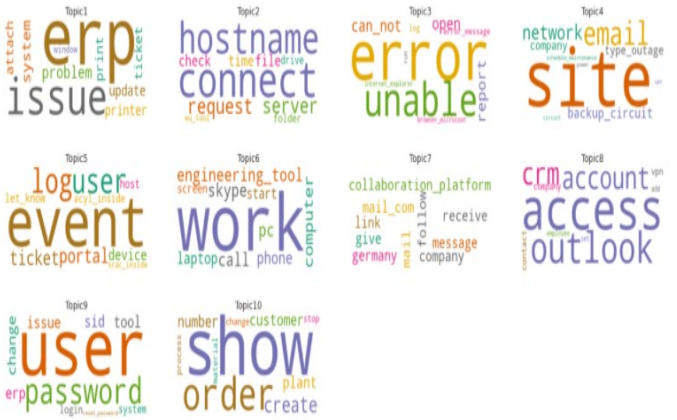
```
num_topics = 10
model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path,
                                              corpus=corpus,
                                              num_topics=num_topics,
                                              id2word=id2word,
                                              random_seed=123,
                                              iterations=100
                                              )
```

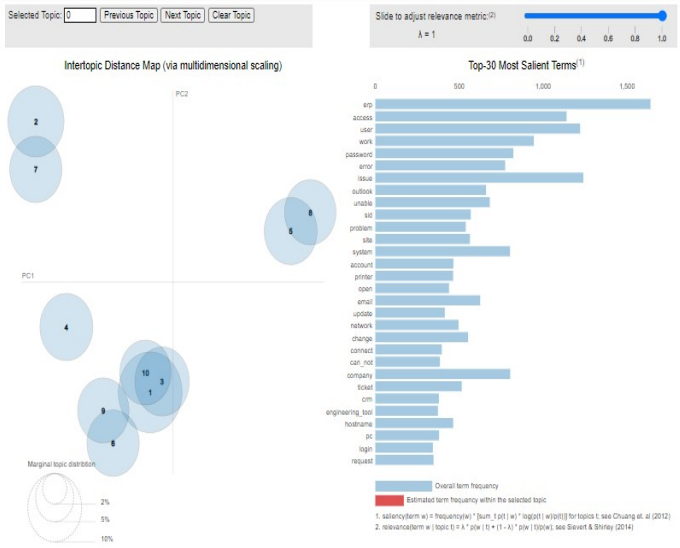
```
Num Topics = 10  LDA Mallet has Coherence Value of 0.5
```

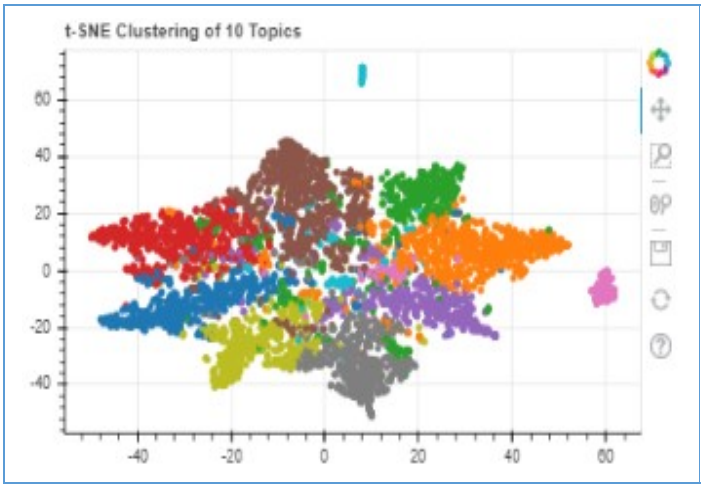
```
num_topics = 10
model_LdaM_bi = gensim.models.wrappers.LdaMallet(mallet_path,
                                                 corpus=corpus_bigram,
                                                 num_topics=num_topics,
                                                 id2word=id2word_bigram,
                                                 random_seed=123,
                                                 iterations=1000,
                                                 )
```

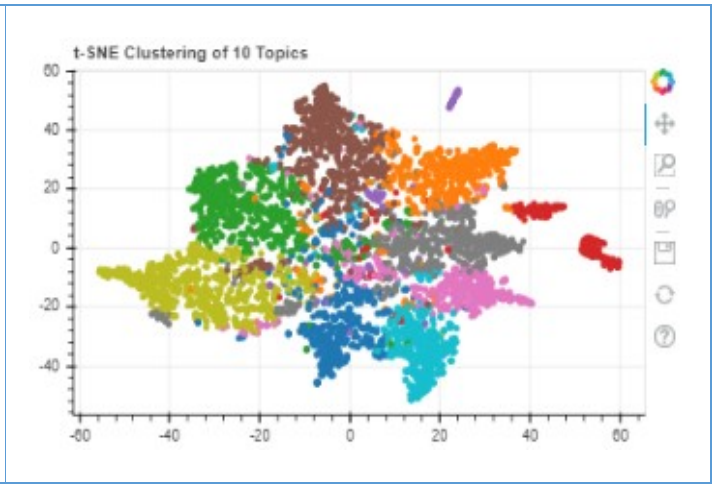
```
Num Topics = 10  LDA Mallet Bigram/BoW has Coherence Value of 0.47
```

INSPECT TOPICS BY LOOKING AT HIGHEST-LIKELIHOOD WORDS IN EACH TOPIC









OBSERVATIONS:

Comparing LDA Mallet Unigram/BoW with LDA Mallet Bigram/Bow with respect to the Evaluating Criteria

1) Coherence Score (c_v) measures: LDA Mallet Unigram/Bow scores 0.5 vs. LDA Mallet Bigram/Bow scores 0.47
2) Inspect topics by looking at highest-likelihood words in each topic:
   a) Topics generated by LDA Mallet Bigram/BoW seems to lack coherence when compared to LDA Mallet Unigram/BoW.
   b) pyLDA vis for LDA Mallet Bigrams/Bow seems to have higher number of overlapped topics
      c. T-SNE cluster plot shows better separation of topics for LDA Mallet Unigrams/BoW in comparison to LDA Mallet Bigrams/BoW

INFERENCE:

Based on above observations, LDA Mallet with Unigrams/BoW is more suitable for the given problem statement

We will proceed to compare LDA Mallet Unigrams/Bag of Words with Unigrams/TFIDF

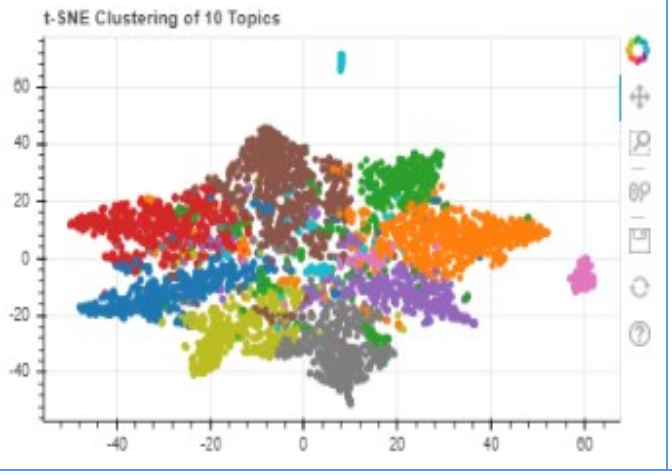
COMPARING: LDA Mallet (Unigrams + BoW) vs. LDA Mallet (Unigrams+ TF-IDF)

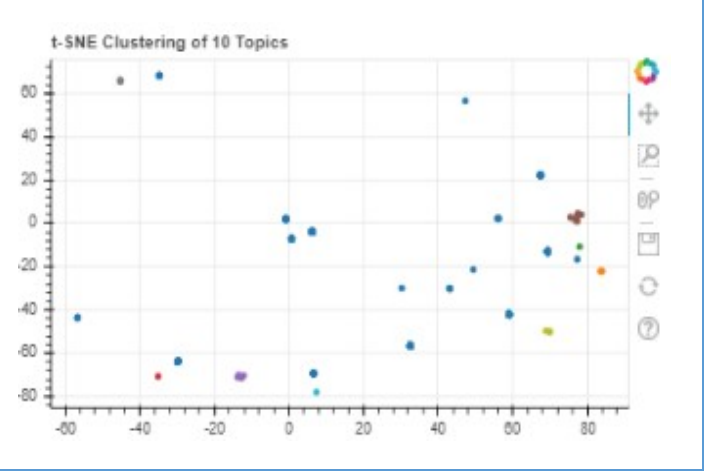
LDA MALLET (UNIGRAM & BAG OF WORDS)	LDA MALLET (UNIGRAMS & TF-IDF)
COHERENCE SCORE	

`num_topics = 10`
`model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path,`
`corpus=corpus,`
`num_topics=num_topics,`
`id2word=id2word,`
`random_seed=123,`
`iterations=100`
`)`
	`num_topics = 10`
	`model_LdaM_tfidf = gensim.models.wrappers.LdaMallet(mallet_path,`
	`corpus=corpus_tfidf,`
	`num_topics=num_topics,`
	`id2word=id2word,`
	`random_seed=123,`
	`iterations=1000,`
	`)`

`Num Topics = 10  LDA Mallet has Coherence Value of 0.5`	`Num Topics = 10  LDA Mallet Unigrams/TFIDF has Coherence Value of 0.`
INSPECT TOPICS BY LOOKING AT HIGHEST-LIKELIHOOD WORDS IN EACH TOPIC	

Topics created are not distinctive using TF-IDF





OBSERVATIONS:

Comparing LDA Mallet Unigram/BoW with LDA Mallet Unigram/TFIDF with respect to the Evaluating Criteria

3) Coherence Score (c_v) measures: LDA Mallet Unigram/Bow scores 0.5 vs. LDA Mallet Unigram/TFIDF scores 0.4
4) Inspect topics by looking at highest-likelihood words in each topic:
   a) Topics generated by LDA Mallet Unigram/TFIDF seems to lack coherence when compared to LDA Mallet Unigram/BoW.
   d. T-SNE cluster plot shows LDA Mallet Unigrams/BoW is significantly better than LDA Mallet Unigram/TFIDF

INFERENCE:

Based on above observations, LDA Mallet with Unigrams/BoW is the optimal model suitable for given problem statement

We will proceed to hypertune LDA Mallet Unigrams/Bag of Words to find the most optimum model

HYPERPARAMETER TUNING OF LDA Mallet (Unigrams + BoW)

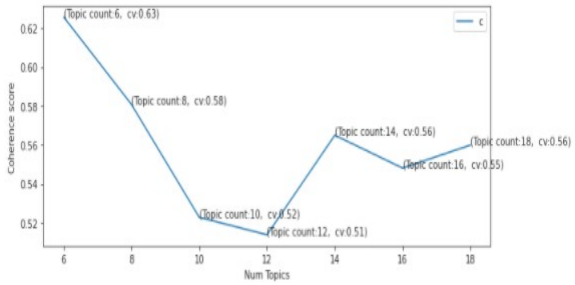
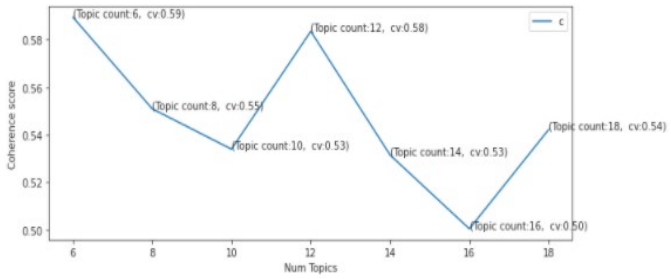
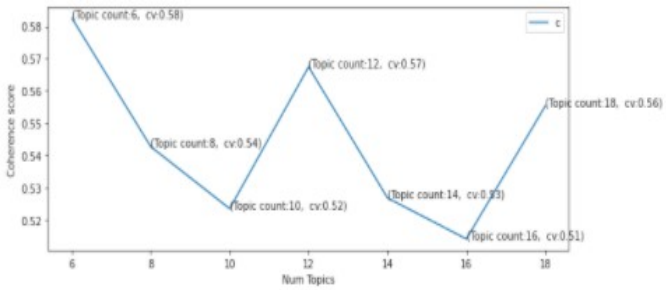
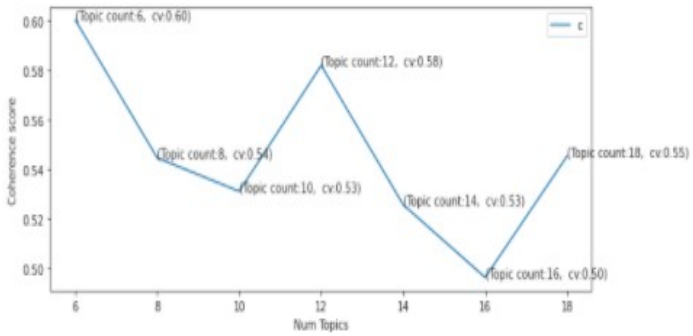
Hyperparameter Optimization parameters provided in http://mallet.cs.umass.edu/topics.php

- **--optimize-interval [NUMBER]** This option turns on hyperparameter optimization, which allows the model to better fit the data by allowing some topics to be more prominent than others. Optimization every 10 iterations is reasonable.
- 'α' alpha **Document-topic Density factor –**

The ‘α’ hyperparameter controls the number of topics expected in the document. Low value of ‘α’ is used to imply that fewer number of topics in the mix is expected and a higher value implies that one would expect the documents to have higher number topics in the mix (default is 50).

- The number of topics to be considered.

```
for num_topics in range(start, limit, step):
    model_LdaM = gensim.models.wrappers.LdaMallet(mallet_path,
                                                  corpus=corpus,
                                                  num_topics=num_topics,
                                                  id2word=dictionary,
                                                  random_seed=1,
                                                  alpha=0.1,
                                                  optimize_interval=10,
                                                  #,topic_threshold=0.20
                                                  )
```

Model 1: Alpha = 0.1, Optimize Interval = 0	Model 2: Alpha = 0.1, Optimize Interval = 10
	
Model 3: Alpha = 0.1, Optimize Interval = 20	**Model 4: Alpha = 0.1, Optimize Interval = 50**
	

OOBSERVATIONS:

- Choosing the number of topics still depends on the business requirement and further qualitative assessment of the keywords within Topics. Number of topics for which average score plateaus, is the sweet spot we are looking for.
- Low value of ‘α’ is used when fewer number of topics in the mix is expected. Based on domain, we expect that each document will have fewer topics. Experiments were performed by changing the values of ‘α’ ranging from 0.1 to 50 and 0.1 was observed to give optimum coherence and hence we are setting α = 0.1.
- Model with Topic 6 although had the highest coherence score but below factors were considered to reject it –
  - **a.** Keywords within this topic were not coherent
  - **b.** 6 topics did not cover the different incident types in the corpus. There were overlaps of incidents over the topics.

- Running experiments with different 'optimize_interval' values, we observe that **Topic Coherence score plateaus** around Topics 8–12
- Model 3 had similar pattern to Model 2 and Model 4 had lower coherence values and hence Model 1 and Model 2 would be considered

INFERENCE:

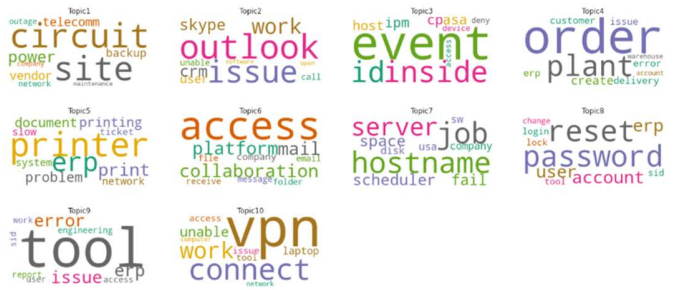
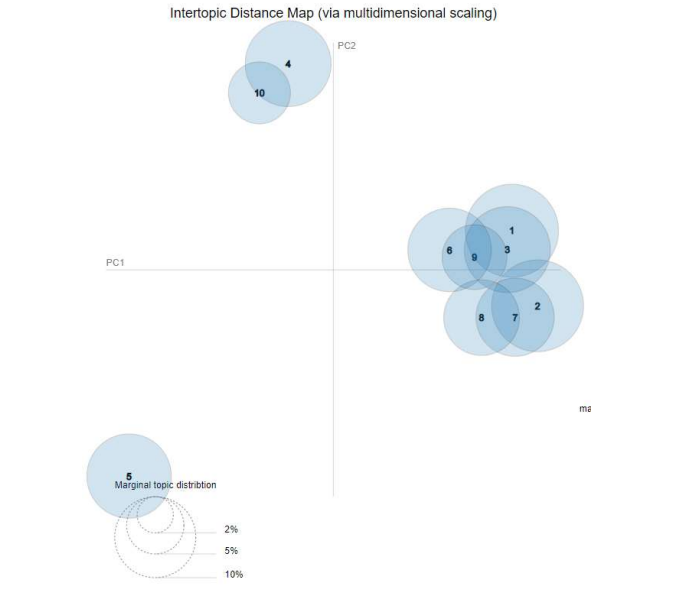
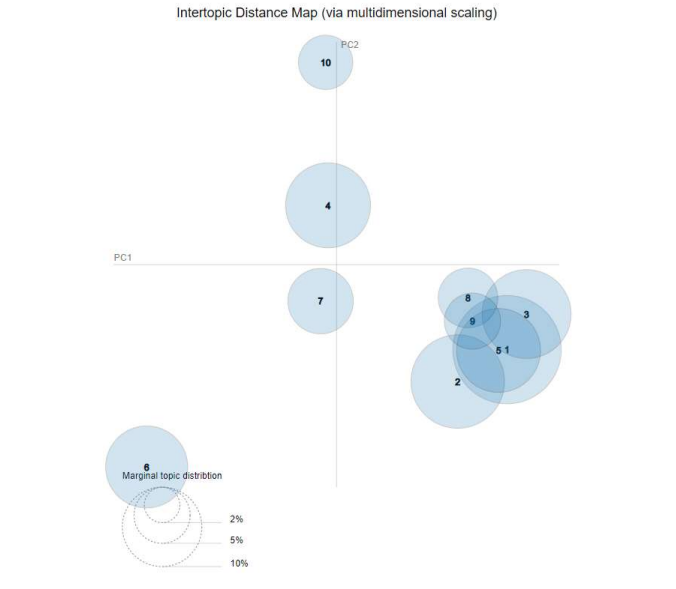
- In absence of supporting information (eg: org structure, cost of resource pool, etc) in problem statement we want to have a balance approach in number of topics chosen, hence **we will pick 10 Topics as our best estimate for Models 1 and 2.**
    a. after inspecting quality of the topics in each model

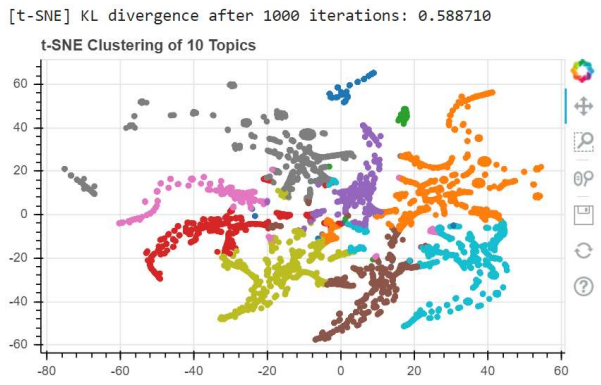
SELECTING OPTIMAL MODEL FOR 10 TOPICS:

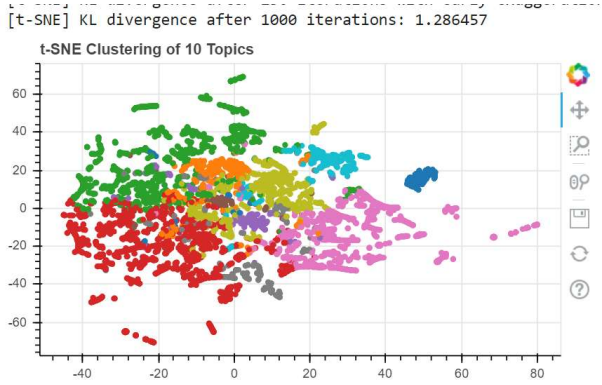
Topic coherence scores :

- Model with Optimize Interval = 0 gives Topic coherence score of 0.52
- Model with Optimize Interval = 10 gives Topic coherence score of 0.53

Key Words generated:

Model 1 with Optimize Interval = 0	Model 2 with Optimize Interval = 10
Word Cloud: Similar keywords and Topic clusters are observed for both models. There is no significant difference	
	
pyLDAvis: In a two dimensional view, it appears that Model 2 has 6 cluster topics overlap versus 7 overlap for Model 1	
	
TSNE Clusters: While similar patern is observed for both models, Model 2 seems to have more denser grouping of words	





Distribution of documents in each Topic: Since LDA is unsupervised training based on latent patterns in the data, topic group size is variable and cannot be compared between models. However we can observe that overall imbalance and range is a significant improvement over original pattern

	Dominant Topic	Doc_Count	Total_Docs_Perc
0	1	161	2.49
1	2	1197	18.49
2	3	74	1.14
3	4	626	9.67
4	5	507	7.83
5	6	714	11.03
6	7	470	7.26
7	8	1048	16.19
8	9	807	12.47
9	10	870	13.44

	Dominant Topic	Doc_Count	Total_Docs_Perc
0	1	176	2.72
1	2	369	5.70
2	3	1375	21.24
3	4	1721	26.58
4	5	184	2.84
5	6	79	1.22
6	7	1233	19.05
7	8	288	4.45
8	9	741	11.45
9	10	308	4.76

Topic assignment quality:

Model with Optimize Interval = 0

	Original_Description	Dominant Topic	Contribution %	Topic Terms
0	verified user details employee name checked user name ad reset password advised user login check caller confirmed able login issue resolved login issue	8	99.55	password, reset, erp, user, account, login, sid, tool, lock, change
1	meetings skype meetings appearing outlook calendar somebody advise correct outlook	2	98.89	outlook, issue, work, skype, crm, user, unable, call, software, open
2	cannot log vpn log vpn	10	84.25	vpn, connect, work, unable, laptop, access, issue, tool, computer, network
3	unable access hr tool page	10	98.24	vpn, connect, work, unable, laptop, access, issue, tool, computer, network
4	skype error	2	58.68	outlook, issue, work, skype, crm, user, unable, call, software, open

Model with Optimize Interval = 10

	Original_Description	Dominant Topic	Contribution %	Topic Terms
	verified user details employee name checked user name ad reset password advised user login check caller confirmed able login issue resolved login issue	7	97.67	password, erp, user, sid, reset, account, login, tool, issue, lock
	meetings skype meetings appearing outlook calendar somebody advise correct outlook	3	94.63	work, skype, issue, outlook, connect, pc, vpn, laptop, user, internet
	cannot log vpn log vpn	3	74.75	work, skype, issue, outlook, connect, pc, vpn, laptop, user, internet
	unable access hr tool page	4	93.97	tool, access, error, crm, engineering, issue, unable, outlook, open, work
	skype error	3	58.57	work, skype, issue, outlook, connect, pc, vpn, laptop, user, internet

OBSERVATIONS:

- Based on above findings, **Model 2 with Optimize_interval = 10 seems to be most optimal**
    a. Coherence score @ 0.53 is comparable with Model 1 score @ 0.52
    b. High Likelihood of keywords within same topics demonstrated by Word Cloud make it easier for human interpretation

#	Keywords	Human Judgement: Group Clusters
Topic1	circuit, site, power, telecomm, vendor, backup, outage, network, company, ticket	Telecom and Networking
Topic2	access, usa, company, drive, hostname, network, folder, sw, server, germany	Country system access
Topic3	work, skype, issue, outlook, connect, pc, vpn, laptop, user, internet	Remote Connectivity
Topic4	tool, access, error, crm, engineering, issue, unable, outlook, open, work	CRM platform
Topic5	message, mail, erp, time, slow, eu, company, tool, germany, problem	Country ERP tool
Topic6	event, inside, imp, id, cp, asa, host, deny, device, source	Device Connectivity
Topic7	password, erp, user, sid, reset, account, login, tool, issue, lock	User account setup
Topic8	printer, print, erp, problem, printing, system, document, order, error, ticket	Printing Issues
Topic9	order, plant, create, tool, erp, error, warehouse, delivery, issue, customer	Warehouse operations
Topic10	job, hostname, server, scheduler, space, fail, agent, sid, disk, consum	Server incidents

    c. Dense TSNE clusters seem to provide better separation supported by 2D, pyLDAvis showing lower Overlap
    d. Dominant Topic assignment seems to be consistent
    e. Concentration of Topics does not seem to create heavy imbalance

INFERENCE:

Newly formed 10 Topics from Model 2 (optimize interval = 10) will be the resultant base to conduct Text Classification using Supervised Learning

APPLY SUPERVISED LEARNING ON 10 TOPICS GENERATED FROM MODEL 2

We are leveraging Topic Modelling for re-classification purposes and condense them into meaningful groups. Supervised learning is run again on this data to compare the performance on the re-classified groups with the original classification.

- Feature Engineering: Combined output of Topic Modelling key words with Original ticket description
- Conducted Multi class Text Classification experiments with below Machine Learning Algorithms

OBSERVATIONS:

1. Text Classification: Test accuracy is averaging < 60%
2. Topic Modelling + Text Classification:
    a. Without Feature Engineering: Test accuracy is averaging > 75% across most of the models with default parameters. Some of the models like SGD is achieving test accuracy of >85%.
    b. With Feature Engineering: Test Accuracy on Topic Modelling output is nearing 100%

Text Classification w/o Topic Modelling Original Description	

	Model	Train_Acc	Test_Acc	F1_Score	Cohen_Kappa	CV_F1
0	SGD_Clf	0.89	0.57	0.56	0.47	0.57
1	MNB_Clf	0.52	0.50	0.65	0.13	0.36
2	RFC_Clf	1.00	0.59	0.67	0.35	0.50
3	linear_SVC	0.86	0.53	0.50	0.44	0.56
4	RBF_SVC	0.74	0.48	0.46	0.38	0.50
5	KNN	1.00	0.60	0.66	0.40	0.53
6	XGB_Clf	0.74	0.58	0.65	0.37	0.53

Topic Modelling + Text Classification

Without Feature Engineering

(Original Description)

	Model	Train_Acc	Test_Acc	F1_Score	Cohen_Kappa	CV_F1
0	SGD_Clf	0.99	0.86	0.86	0.84	0.86
1	MNB_Clf	0.85	0.78	0.80	0.73	0.75
2	RFC_Clf	1.00	0.79	0.79	0.74	0.78
3	linear_SVC	0.99	0.87	0.87	0.85	0.87
4	RBF_SVC	0.98	0.86	0.86	0.83	0.84
5	KNN	1.00	0.80	0.80	0.76	0.78
6	XGB_Clf	0.84	0.75	0.75	0.70	0.77

Topic Modelling + Text Classification

With Feature Engineering

(Topic Model Keywords + Original Description)

	Model	Train_Acc	Test_Acc	F1_Score	Cohen_Kappa	CV_F1
0	SGD_Clf	1.00	1.00	1.00	1.00	1.00
1	MNB_Clf	0.97	0.96	0.96	0.95	0.94
2	RFC_Clf	1.00	1.00	1.00	1.00	1.00
3	linear_SVC	1.00	1.00	1.00	1.00	1.00
4	RBF_SVC	1.00	1.00	1.00	1.00	0.99
5	KNN	1.00	0.96	0.96	0.95	0.95
6	XGB_Clf	1.00	1.00	1.00	1.00	1.00

3. Inspecting Test (1295 rows) Classification Report for SGD classifiers shows favorable accuracy results across all classes and improvement with Topic Modelling:

Only Text Classification (73 groups): 55% >>Topic Modelling (10 Groups) + Text Classification: 85% (w/o Feature engineering >> Topic Modelling (10 Groups) + Text Classification: 99.8% (with Feature engineering

RandomSearchCV applied on SGD classifier:

```
converted the input data into vector form

Results with RandomizedSearchCV
{'alpha': 1e-05, 'average': False, 'class_weight': 'balanced',
'early_stopping': False, 'epsilon': 0.1, 'eta0': 5.0, 'fit_intercept': True,
'l1_ratio': 0.15, 'learning_rate': 'adaptive', 'loss': 'log', 'max_iter':
2000, 'n_iter_no_change': 5, 'n_jobs': None, 'penalty': 'l2', 'power_t': 0.5,
'random_state': 13, 'shuffle': True, 'tol': 0.001, 'validation_fraction':
0.1, 'verbose': 0, 'warm_start': False}

test score 0.7637065637065638

train score 0.9895732766943426
-------------------------------------------
```

Text Classification w/o Topic Modelling Original Description – 73 GROUPS	Topic Modelling + Text Classification Without Feature Engineering (Original Description) – 10 GROUPS

	precision	recall	f1-score	support
0	0.814000	0.689831	0.746789	590.00000
1	0.000000	0.000000	0.000000	1.00000
2	0.272727	0.400000	0.324324	15.00000
3	0.250000	0.166667	0.200000	6.00000
4	0.397059	0.540000	0.457627	50.00000
...	...	...	...	...
71	0.830189	0.771930	0.800000	57.00000
72	0.421053	0.470588	0.444444	17.00000
accuracy	0.568340	0.568340	0.568340	0.56834
macro avg	0.345219	0.317591	0.314335	1295.00000
weighted avg	0.604956	0.568340	0.575681	1295.00000

	precision	recall	f1-score	support
0	1.000000	0.937500	0.967742	32.00000
1	0.882979	0.846939	0.864583	98.00000
2	0.833866	0.915789	0.872910	285.00000
3	0.876623	0.846395	0.861244	319.00000
4	0.760000	0.808511	0.783505	47.00000
5	1.000000	0.937500	0.967742	16.00000
6	0.907080	0.907080	0.907080	226.00000
7	0.826087	0.850746	0.838235	67.00000
8	0.907563	0.818182	0.860558	132.00000
9	0.971831	0.945205	0.958333	73.00000
accuracy	0.877220	0.877220	0.877220	0.87722
macro avg	0.896603	0.881385	0.888193	1295.00000
weighted avg	0.879256	0.877220	0.877402	1295.00000

COHEN KAPPA BENCHMARKING: Cohen suggested the Kappa result be interpreted as follows: values ≤ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement

Based on above benchmarking practice, SGD classifier Topic Modelling + Text Classification (on original description) is giving 81+% Cohen Kappa score which is termed as almost perfect agreement

```
----------------------------------------------------------
Training accuracy: 99.363%
----------------------------------------------------------
Testing accuracy: 86.795%
----------------------------------------------------------
cohen_kappa score: 84.383%
----------------------------------------------------------
cross validation f1 weighted score: 0.8462011484092107
----------------------------------------------------------
```

INFERENCE:

- Significant improvement in accuracy scores observed when Topic Modelling is performed prior to Text Classification
- Topic Modelling has attributed to correcting significant class imbalance by using latent (hidden) patterns in the ticket description for appropriate grouping structure of 10 Groups
- **Test Accuracy scores is averaging > 80% (Without feature engineering)**
  - **Exceeds Benchmark:** Linear SVC classifier and SGD with Topic Modelling + Text Classification (on original description) is giving ~85% cross validation (cv) accuracy score and >0.84 Cohen Kappa score which is termed as almost perfect agreement. SGD is giving slightly better accuracy score compared to LinearSVC.
- **Test Accuracy is ~100% (With Feature engineering)** across all experiments beating manual classification score of 75% accuracy as given in the problem statement

CONFIDENCE INTERVAL

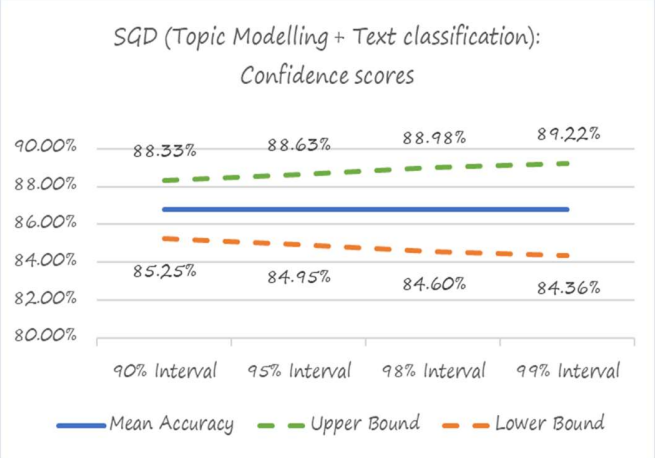
- In general, the confidence interval for classification accuracy can be calculated as follows:
  - **accuracy +/- const * sqrt((accuracy * (1 — accuracy)) / n)**
- Mean Classification accuracy for Test set for SGD Classifier achieved from Topic Modelling + Text Classification (on original incident description) is 88.03%.
- The values for "const" are provided from statistics, and common values used are:

1.64 (90%)	1.96 (95%)	2.33 (98%)	2.58 (99%)

- "n" is the number of observations / records from Test Data which in this case is 1297
- Confidence interval estimates for Test Data Set of 1295 records with Mean classification accuracy of 87%:

INFERENCE:

- We expect Test accuracy for SGD classification to range
  - between 85.25% - 88.33% @ 90% Confidence interval
  - between 84.95% - 88.63% @ 95% Confidence interval
  - between 84.60% - 88.98% @ 98% Confidence interval
  - between 84.36% - 89.22% @ 99% Confidence interval



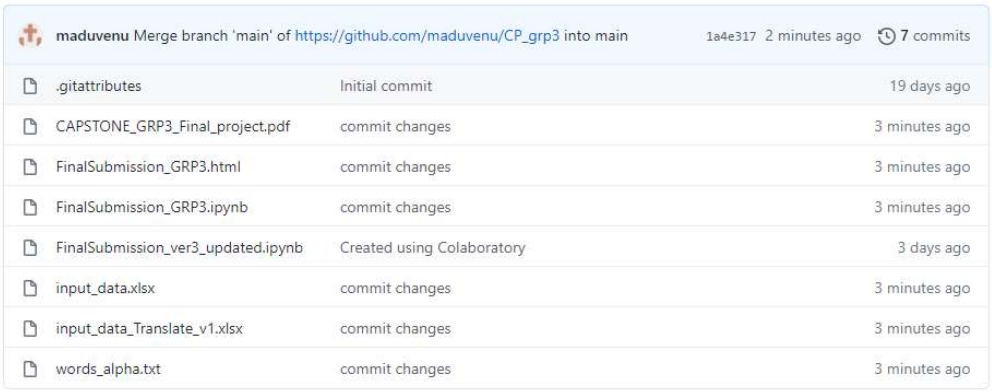
DEPLOYMENT:

Topic Modelling generally takes more time and resources to execute. Running this model in a live environment would turn out to be costlier and less effective and may not be a recommended approach for deployment.

To overcome, we experimented and ran Supervised learning techniques on the outcome/themes generated from Topic Modelling. The accuracy of test data is significantly good with almost all the supervised models that were run.

Based on our hypothesis, EDA, learning and various experiments conducted during the course of this project we recommend to deploy **SGD Supervised model** in the live environments to classify the tickets in their respective re-classified themes/groups generated and finalized as an outcome of Topic Modelling.

Entire code has been checked in along with reference files in github : https://github.com/maduvenu/CP_grp3



IMPLICATIONS:

EFFECT OF OUR SOLUTION IN THE DOMAIN:

- Based on the model simulation the proposed solution exceeds the accuracy of the existing classification (75%) by more than 10% and also is in "perfect range" of cohen-kappa benchmark.
- The new reclassified groups are more cohesive and aligned to few key topics/themes.
- In the near term the reclassification of the groups and implementing of the solution may lead to additional restructuring cost to the organization. Depending on the existing systems, the "Incident management system" may need to be updated to align with the proposed solution. It may also require retraining of the support team at L1/L2/L3 to handle the new groups.
- In the longer term, the proposed solution is expected bring in overall cost reduction as the number of staff required to assign the tickets at L1 level can be gradually reduced or reassigned to other levels.
- As the model simulation show >85% accuracy, it will help the company to reduce TAT for ticket resolution thereby increasing productivity and providing indirect cost benefits to the company. Lower TAT also keeps overall morale of the workforce high.
- The high-level themes/groups provide an opportunity to the company to focus on its core business and outsource some of the groups/incidents to 3rd party. This can further help the company to reduce the various system downtime as the groups will now be addressed by an experienced/expert team.
- Themes based on the topics generated can be used by the organization to classify the tickets as is done within the scope of project.
- We leveraged Topic Modelling for re-classification purposes of the groups and come up with meaningful themes. Supervised learning was run again on this data to compare the performance on the re-classified groups with the original classification. Topic modelling generally takes a significant amount of computation time and resources as compared to other supervised models.
- Based on the above points, in the longer term the benefits could significantly outweigh the short-term restructuring cost and challenges to the company.

LIMITATIONS:

LIMITATIONS IN OUR SOLUTION:

- This model was trained on limited data points so the performance of the model could be different between the training/validation and in the actual production environment.

- Some more work may be needed to deploy the actual model e.g. implementing the UI interface, integration with cloud infrastructure to name a few.
- Proposed model handles incidents only in English language as language translation is a paid service. If other languages need to be supported, then the inline language translation need to be integrated to the model or other solutions need to be explored. Based on this the model re tuning may be needed and performance may vary.

LIMITATIONS TO OUR SOLUTION IN REAL WORLD:

- As the solution was implemented on a standalone basis without any interaction with organization and their SME, the solution may need some further fine tuning to better adapt it to the organization requirements.
- Depending on the organization size and structure, there could challenges to change the existing process of reclassifying the groups as the systems are generally heritage in nature and may have a large volume of data to be transformed. Due to this actual cost benefit may slightly differ than what has been inferred from the model performance.
- Topic modelling does not seem to be widely used to re classify the existing groups and a standard industry practice.
- GSDMM is still an area of academic research and not much support is available in the standard libraries.

REFLECTIONS:

LEARNINGS:

There were multiple areas/points/approach in which we had a significant amount of learning –

- EDA – the entire solution was based on how this was carried out. During the project, each member had a set of findings to be shared and there were numerous insights.
- Text Pre-processing – Multiple API's were checked, and their results tested. We had to spend a good amount of time to understand which one brings in more value on the table. Below are a few that should be mentioned:
    - Language Detection
    - Spellcheck
- Optimizing the topic modelling

Topic modeling technique and knowledge around it was not covered as part of the current AIML course.

A great deal of effort and significant amount of time went in to learn about Topic Modeling in general, its application in various cases, different measures of comparison and its actual implementations in the real world. We did a great amount of research on how various models have been built and tested for multiple text classification purposes. As the approach to reclassify the groups using topic modelling is not widely used in the industry, it took some time for the team to get convinced that this approach could be for this problem. We had to start from scratch, learn, unlearn and relearn on how various models have been built and tested it on the problem before finalizing the approach. There was also a tremendous amount of time spent to learn various visualization techniques and its applicability along with various topic model evaluation techniques. During these 6 weeks of project, based on the research and learning, there was an excellent alignment within the team and our mentor. We feel satisfied that we have been able to push us ourselves beyond what we have learnt in the course and come with a different approach to solve this problem.

We would recommend the faculty and the course coordinators to include this topic in future AIML courses.

WHAT COULD BE DONE DIFFERENTLY?

We spent a lot of time to discuss and brainstorm on the solution and the model that we wanted to choose and execute. The regular classification techniques were run at a later duration of the course of the project.

Looking back, model deployment is one of the areas where were would have liked to spend more time but couldn't do so.

REFERENCES:

- https://www.researchgate.net/publication/334667298_Topic_Modeling_A_Comprehensive_Review
- https://radimrehurek.com/gensim/
- https://nlpforhackers.io/topic-modeling/
- http://mallet.cs.umass.edu/topics.php
- https://topicmodels.west.uni-koblenz.de/
- https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/
- https://medium.com/@lettier/how-does-lda-work-ill-explain-using-emoji-108abf40fa7d
- https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0
- https://towardsdatascience.com/short-text-topic-modeling-70e50a57c883