AUTOMATIC TICKET ASSIGNMENT

CAPSTONE PROJECT – FINAL REPORT

Submitted by:

Rajesh.M

Renato Falcon Lyke

Hema Medicharla

Somnath Mohapatra

D Mallikarjuna Reddy

Lingaraj sabat

Mentored by:

Venkatesha K H

Contents

Summary of the Problem Statement, Data and Findings

Problem statement

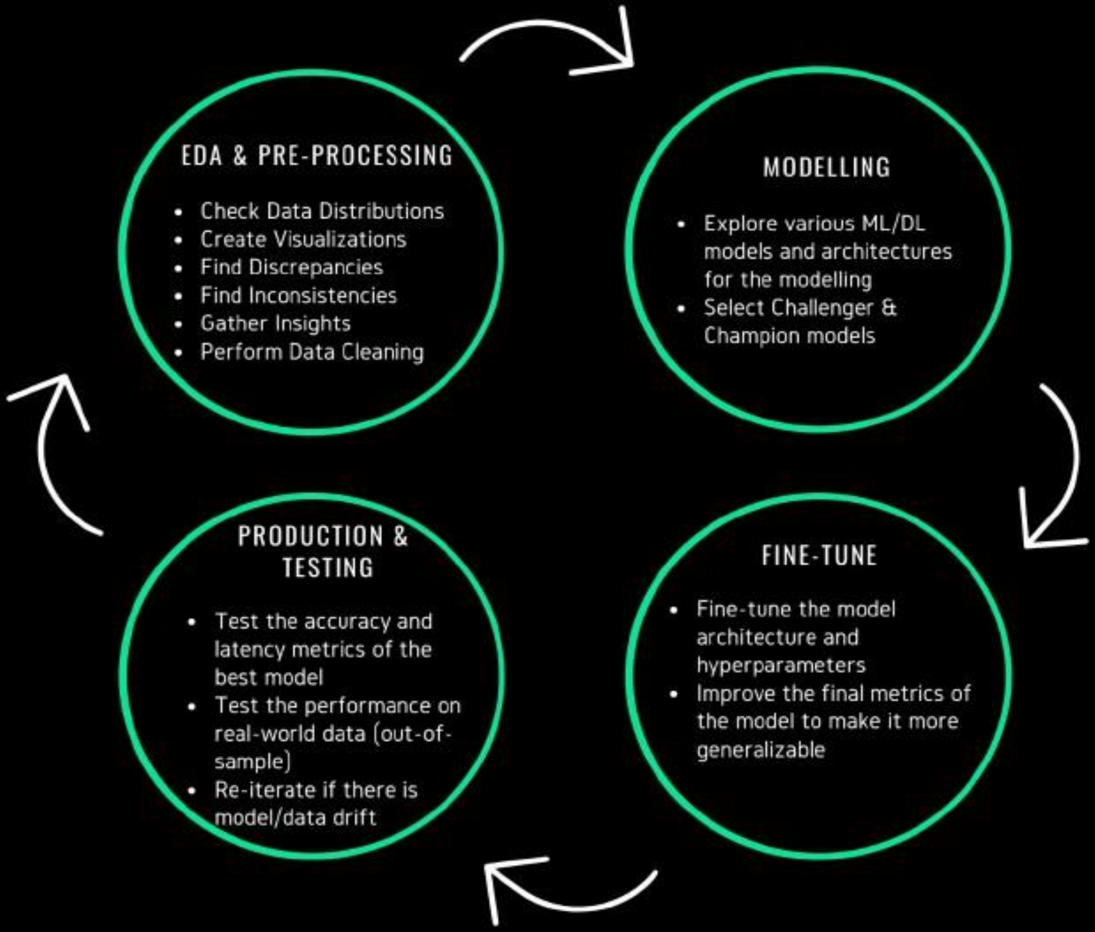
Excellent & Effective Customer Support is Quintessential to the running of any business organization, no matter its size—84% of organizations working to improve customer service report an increase in revenue. In the current scenario, various incidents faced by the business are all assigned to two L1/L2 teams. Only 54% of these incidents can be resolved at this level. For all the rest, the incidents are escalated to L3 teams to be resolved. Additionally, the manual reassignment to various functional groups was found to have an error rate of around 25%. This added overhead cost of time and resources of re-assigning the incidents is detrimental to the customer support efficiency causing delays and bad customer experiences. A better allocation and practical usage of the functional groups' resources will result in substantial cost, time savings and better customer support overall.

Hence, we aim to build a classifier using state-of-the-art Machine learning algorithm to classify the tickets to various functional groups by just analysing the text of the various issues, thereby driving direct business value in IT customer support.

Automatic Ticket CLassification

PROJECT LIFE-CYCLE

This is the general life-cycle of any AI/ML project that we are following in our Capstone project; We go through each step of the analysis to inform and drive the next stage of modelling and re-iterate till we are satisfied on the performance.



EDA & PRE-PROCESSING

- Check Data Distributions
- Create Visualizations
- Find Discrepancies
- Find Inconsistencies
- Gather Insights
- Perform Data Cleaning

MODELLING

- Explore various ML/DL models and architectures for the modelling
- Select Challenger & Champion models

FINE-TUNE

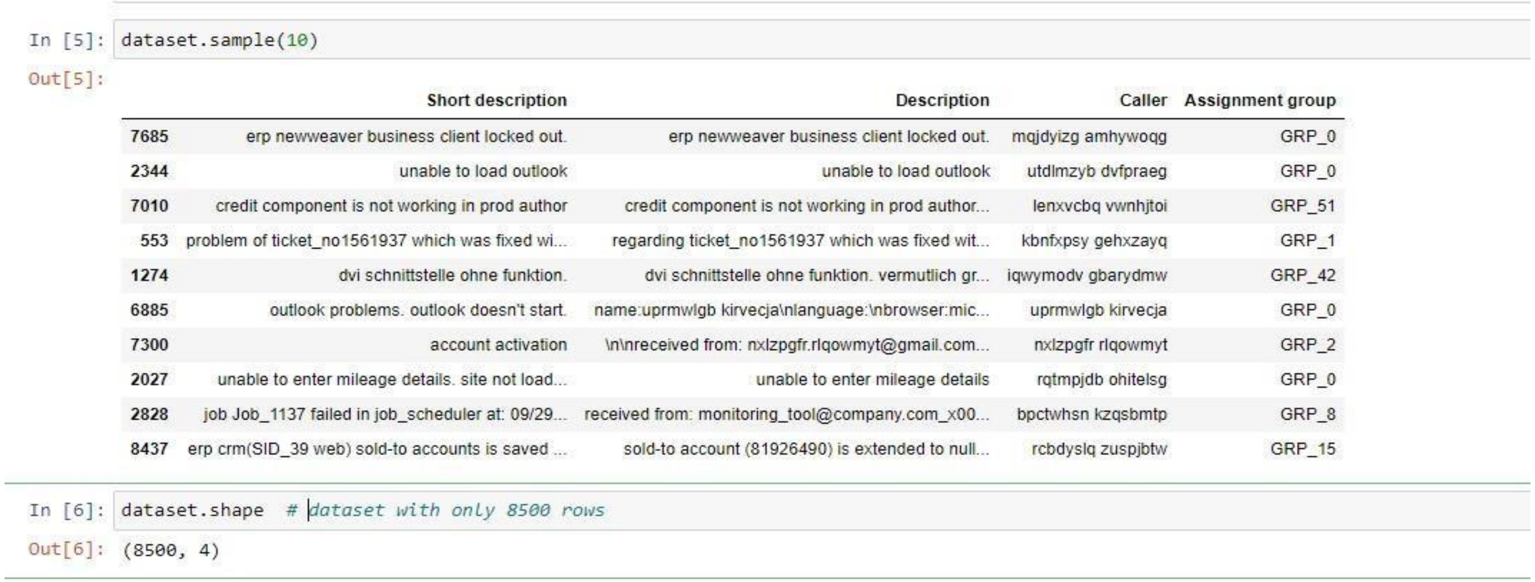
- Fine-tune the model architecture and hyperparameters
- Improve the final metrics of the model to make it more generalizable

PRODUCTION & TESTING

- Test the accuracy and latency metrics of the best model
- Test the performance on real-world data (out-of-sample)
- Re-iterate if there is model/data drift

Dataset

Our dataset consists of 8500 data points each consisting of a short description of the issue, a longer description, the caller name (appears to be encrypted and anonymized in the given dataset to protect privacy) and the target class group to which the issue has to be assigned to.

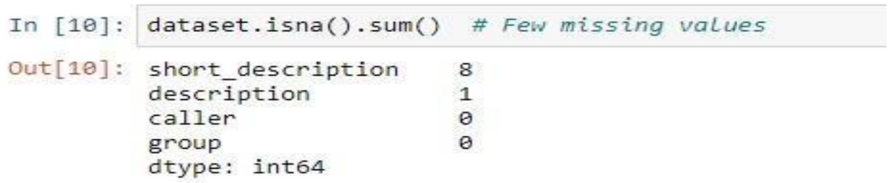


```
In [5]: dataset.sample(10)
Out[5]:
```

	Short description	Description	Caller	Assignment group
7685	erp newweaver business client locked out.	erp newweaver business client locked out.	mqjdyizg amhywoqg	GRP_0
2344	unable to load outlook	unable to load outlook	utdlmzyb dvfpraeg	GRP_0
7010	credit component is not working in prod author	credit component is not working in prod author...	lenxvcbq vwnhjtoi	GRP_51
553	problem of ticket_no1561937 which was fixed wi...	regarding ticket_no1561937 which was fixed wit...	kbnfxpsy gehxzayq	GRP_1
1274	dvi schnittstelle ohne funktion.	dvi schnittstelle ohne funktion. vermutlich gr...	iqwymodv gbarydmw	GRP_42
6885	outlook problems. outlook doesn't start.	name:uprmwlgb kirvecja\nlanguage:\nbrowser:mic...	uprmwlgb kirvecja	GRP_0
7300	account activation	\n\nreceived from: nxlzpgfr.rlqowmyt@gmail.com...	nxlzpgfr rlqowmyt	GRP_2
2027	unable to enter mileage details. site not load...	unable to enter mileage details	rqtmpjdb ohitelsg	GRP_0
2828	job Job_1137 failed in job_scheduler at: 09/29...	received from: monitoring_tool@company.com_x00...	bpctwhsn kzqsbmtp	GRP_8
8437	erp crm(SID_39 web) sold-to accounts is saved ...	sold-to account (81926490) is extended to null...	rcbdyslq zuspjbtw	GRP_15

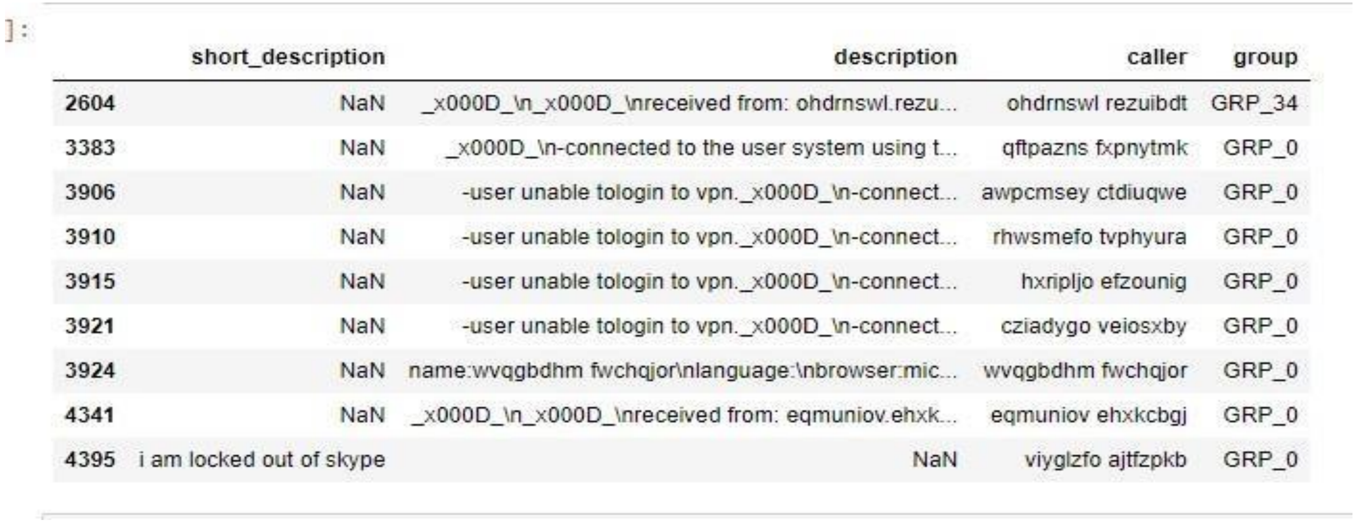
```
In [6]: dataset.shape  # dataset with only 8500 rows
Out[6]: (8500, 4)
```

There seem to be missing values in the Short description and Description columns, which needs to be looked into and handled. There are 8 nulls/missing values present in the Short description and 1 null/missing value present in the description column.



```
In [10]: dataset.isna().sum()  # Few missing values

Out[10]: short_description    8
         description          1
         caller               0
         group                0
         dtype: int64
```

The independent features are short descriptions and descriptions and the target/dependent feature is the group.

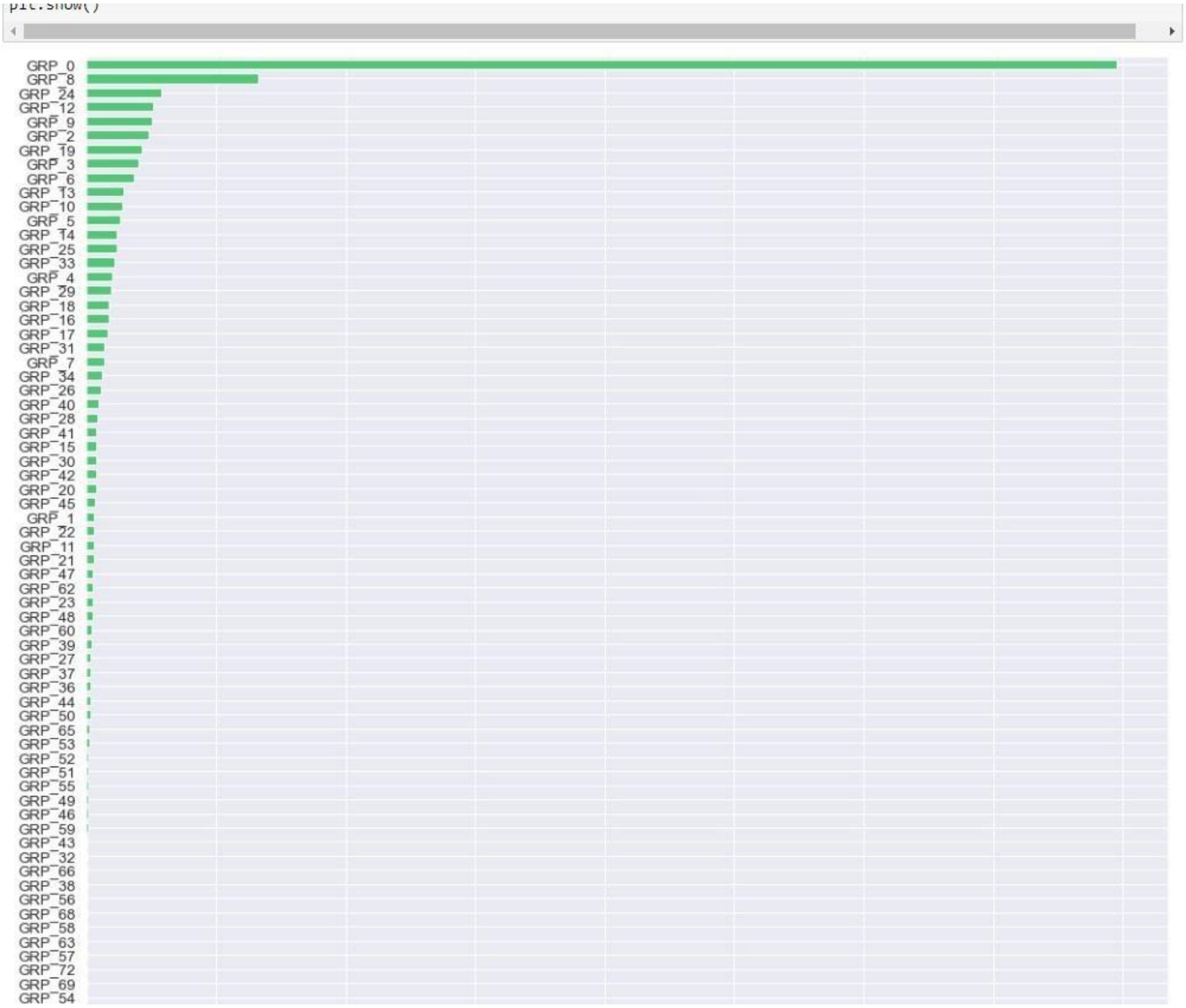


]:

	short_description	description	caller	group
2604	NaN	_x000D_\n_x000D_\nreceived from: ohdrnswl.rezu...	ohdrnswl rezuibdt	GRP_34
3383	NaN	_x000D_\n-connected to the user system using t...	qftpazns fxpnytmk	GRP_0
3906	NaN	-user unable tologin to vpn._x000D_\n-connect...	awpcmsey ctdiuqwe	GRP_0
3910	NaN	-user unable tologin to vpn._x000D_\n-connect...	rhwsmefo tvphyura	GRP_0
3915	NaN	-user unable tologin to vpn._x000D_\n-connect...	hxripljo efzounig	GRP_0
3921	NaN	-user unable tologin to vpn._x000D_\n-connect...	cziadygo veiosxby	GRP_0
3924	NaN	name:wvqgbdhm fwchqjor\nlanguage:\nbrowser:mic...	wvqgbdhm fwchqjor	GRP_0
4341	NaN	_x000D_\n_x000D_\nreceived from: eqmuniov.ehxk...	eqmuniov ehxkcbgj	GRP_0
4395	i am locked out of skype	NaN	viyglzfo ajtfzpkb	GRP_0

Summary of the approach to EDA and Pre-Processing

Target Distribution



```
plt.show()
```

The Target class distribution is extremely skewed and heavily imbalanced as the majority of incidents are from Group 0 followed by Group 8, 24, 12, 9, 2 and we find an imbalanced dataset for the rest of the groups.

A large no. of entries for “Group 0” which account for ~47% of the data and remaining are grouped together as “Other” as there is not much information with the groups individually.

Choosing a Metric

This is a multi-class classification problem, where the machine learning model will try to predict if each row is one of the 74 possibilities.
The majority class is GRP_0, which occurs in 46.78% of the observations.
The most common metrics for a multi-class classification problem are AUC, F1-score and accuracy.
Accuracy is not suitable for an imbalanced classification problem.

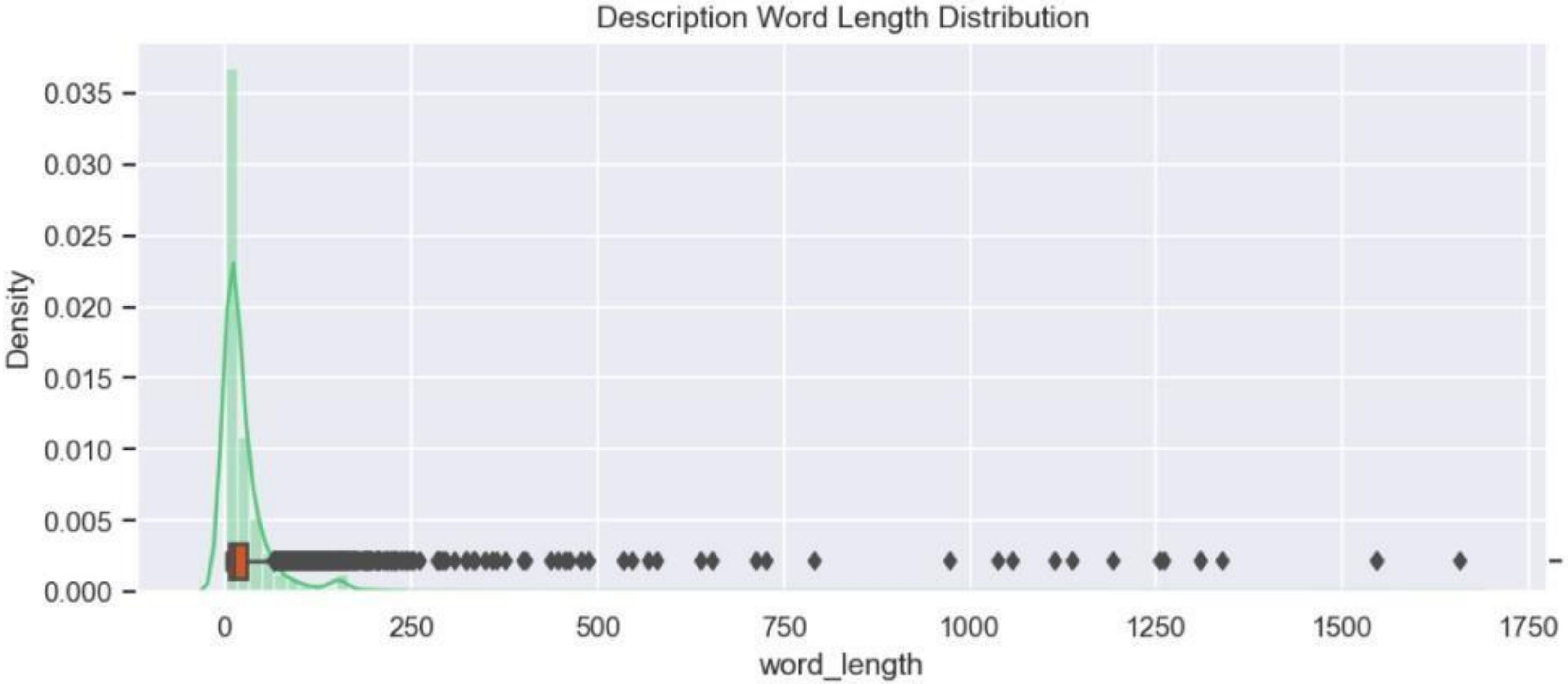
Language Detection

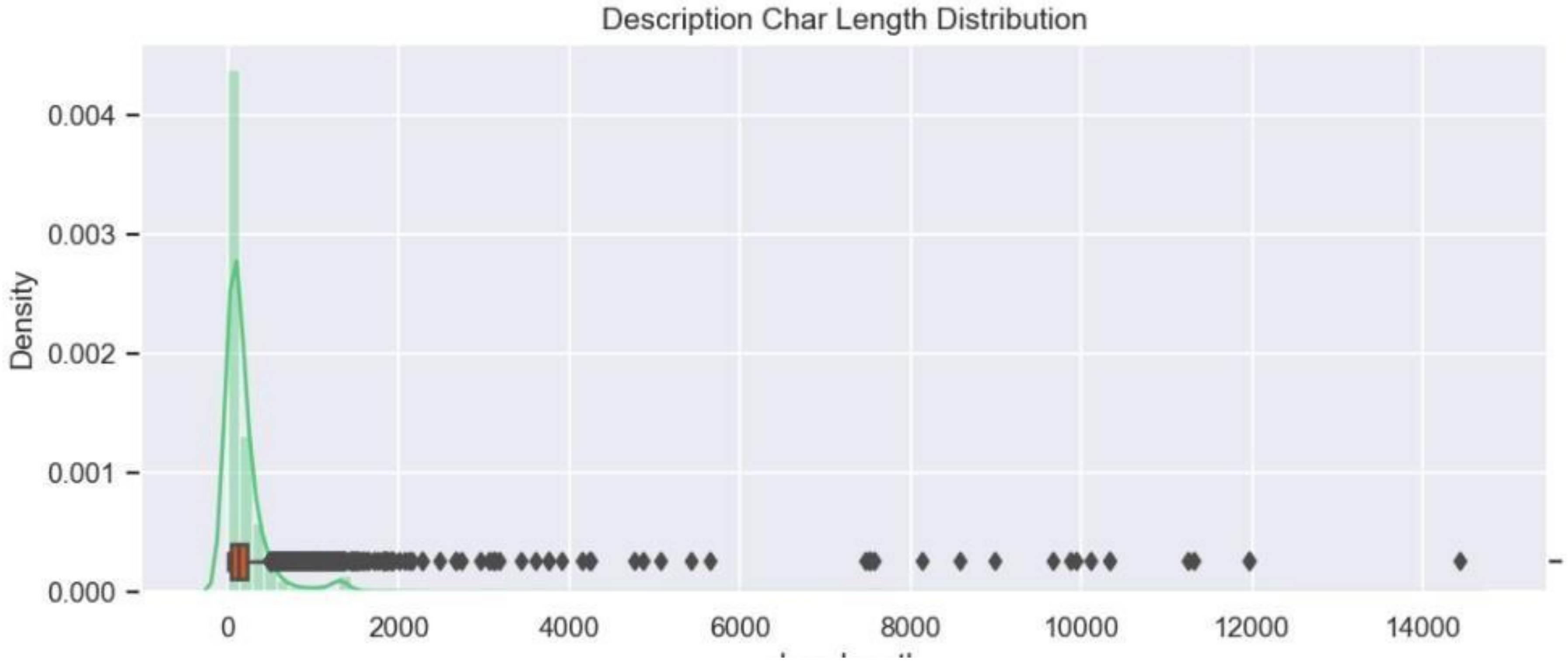
We have seen around 17.03 % of the foreign language texts in the description. These texts include German, French, Chinese, Italian and other European languages in relatively small numbers.

Keyword Extraction

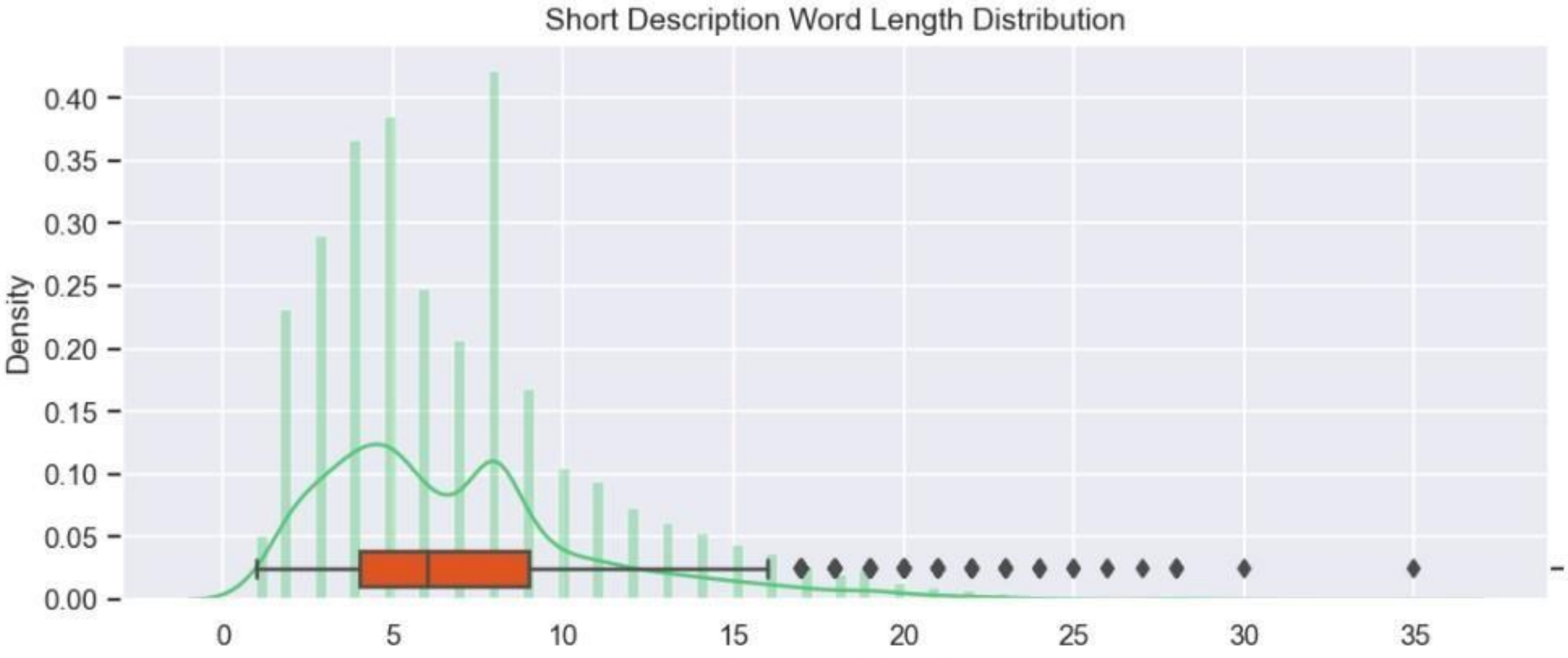
YAKE is a lightweight unsupervised automatic keyword extraction using an unsupervised approach that rests on text statistical features extracted from single documents to select the most important keywords of a text and is independent of corpus, domain and language.

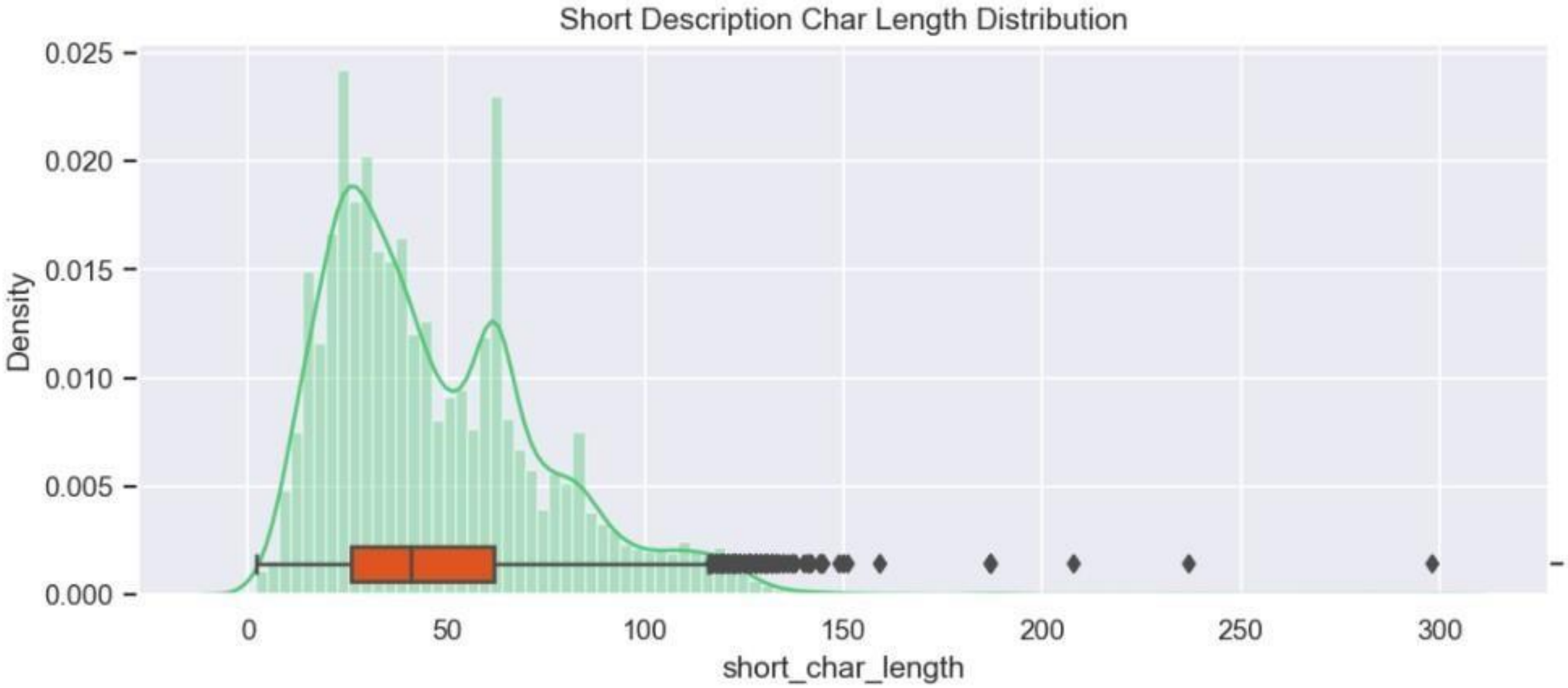
Description Word and Character Counts distribution





Short Description Word and Character Counts Distribution

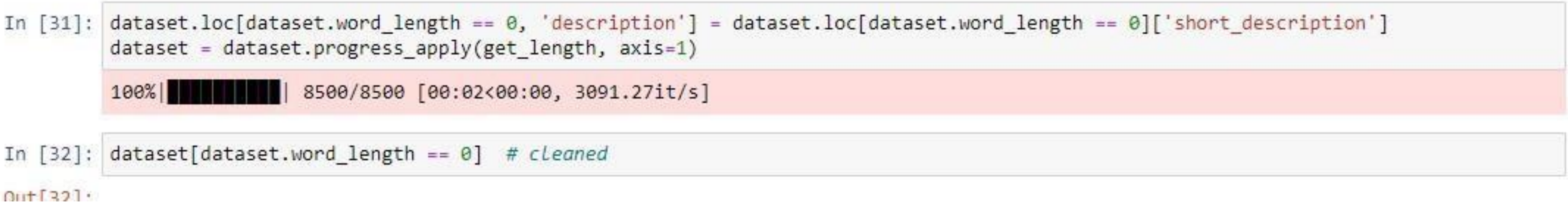

Short Description Word Length Distribution
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
35



Inconsistencies & Discrepancies found during EDA

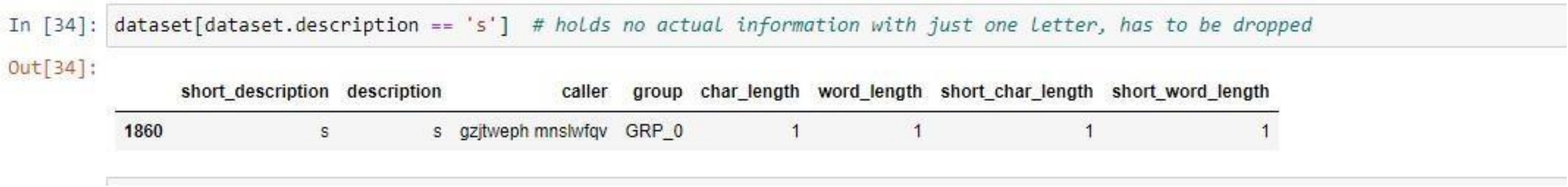
Clean up the unwanted information from initial observations. Imputing the dataset which has no data, one and two-word length by their corresponding short description.

No word length ⇒ Imputed the description with the corresponding short description



```
In [31]: dataset.loc[dataset.word_length == 0, 'description'] = dataset.loc[dataset.word_length == 0]['short_description']
         dataset = dataset.progress_apply(get_length, axis=1)

         100%|██████████| 8500/8500 [00:02<00:00, 3091.27it/s]

In [32]: dataset[dataset.word_length == 0]  # cleaned

Out[32]:
```

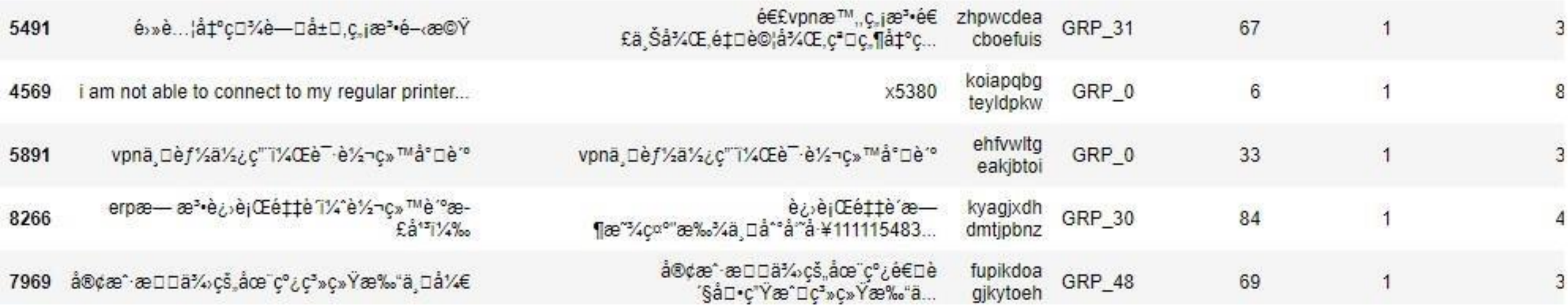
One word length ⇒ Drop the row as the descriptions have no discernible information



```
In [34]: dataset[dataset.description == 's']  # holds no actual information with just one letter, has to be dropped

Out[34]:
```

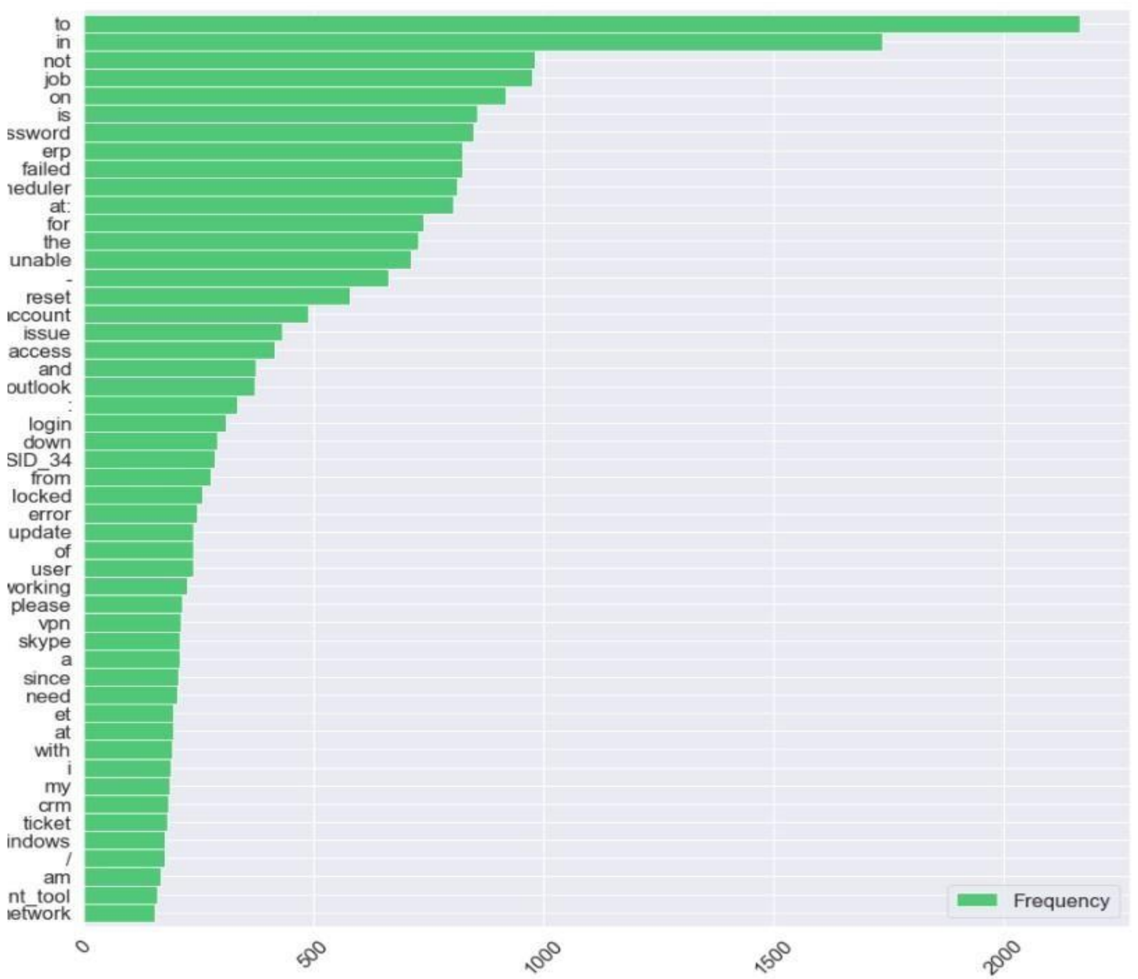
	short_description	description	caller	group	char_length	word_length	short_char_length	short_word_length
1860	s	s	gzjtweph mnslwfqv	GRP_0	1	1	1	1

Fix the encoding



5491	é›»è…¦å‡ºç□¾è—□å±□,ç„¡æ³•é–‹æ©Ÿ	é€£vpnæ™‚,ç„¡æ³•é€£ä¸Šå¾Œ,é‡□è©¦å¾Œ,ç³□ç„¶å‡ºç...	zhpwcdea cboefuis	GRP_31	67	1	3
4569	i am not able to connect to my regular printer...	x5380	koiapqbg teyldpkw	GRP_0	6	1	8
5891	vpnä¸□èƒ½ä½¿ç”¨ï¼Œè¯·è½¬ç»™å°□è´º	vpnä¸□èƒ½ä½¿ç”¨ï¼Œè¯·è½¬ç»™å°□è´º	ehfvwltg eakjbtoi	GRP_0	33	1	3
8266	erpæ— æ³•è¿›è¡Œé‡‡è´ï¼ˆè½¬ç»™è´ºæ-£å¹³ï¼‰	è¿›è¡Œé‡‡è´æ—¶æ˜¾ç¤º"æ‰¾ä¸□åˆ°å¯¹å·¥111115483...	kyagjxdh dmtjpbnz	GRP_30	84	1	4
7969	å®¢æˆ·æ□□ä¾›çš„åœ¨çº¿ç³»ç»Ÿæ‰“ä¸□å¼€	å®¢æˆ·æ□□ä¾›çš„åœ¨çº¿é€□è´§å□•ç"Ÿæˆ□ç³»ç»Ÿæ‰"ä...	fupikdoa gjkytoeh	GRP_48	69	1	3

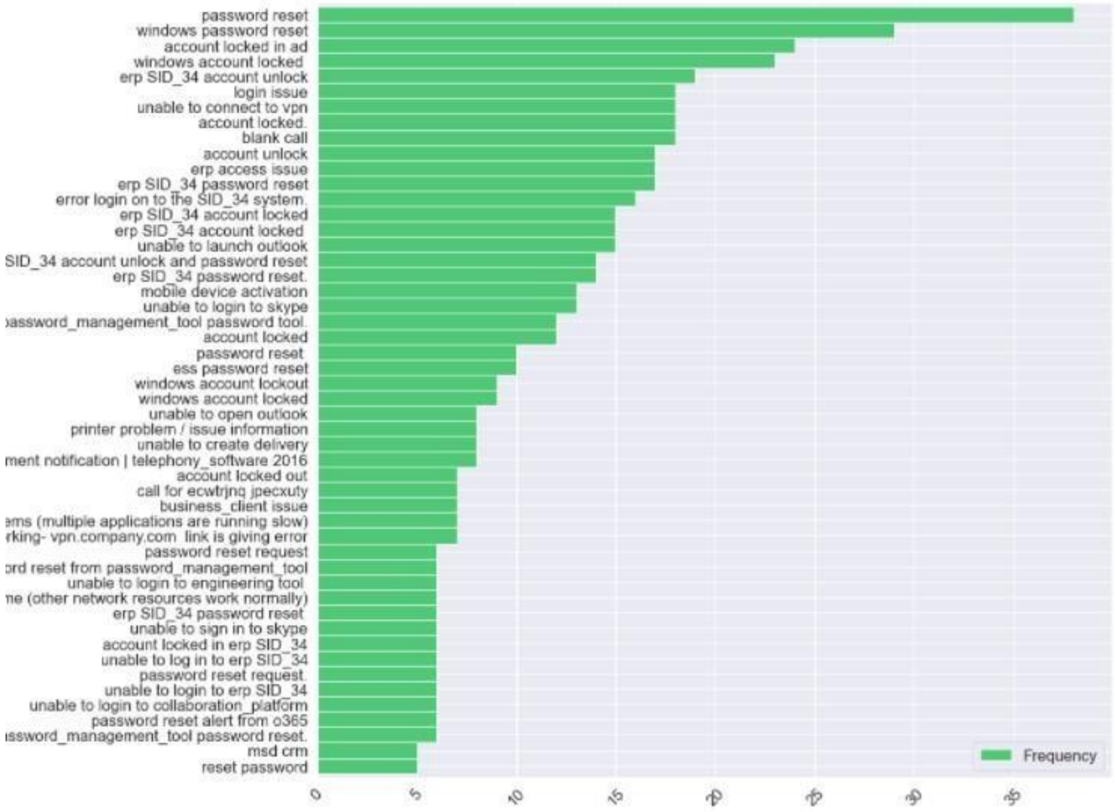
Word Frequency Distributions



We have observed that words like "to", "the", "in".. etc., are occurring most frequently in the descriptions. These words will not add any predictive power to the models and will need to be removed as part of the stopword removal process during data pre-processing.

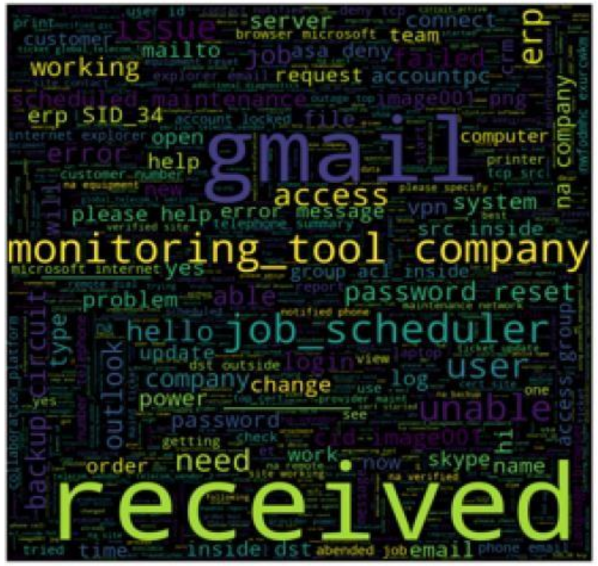
Also, anchor words like "from:" and "received" and email addresses, punctuations, numbers are also occurring relatively frequently. We will remove these as well as part of the pre-processing

Word Frequency Distribution after Data Cleaning

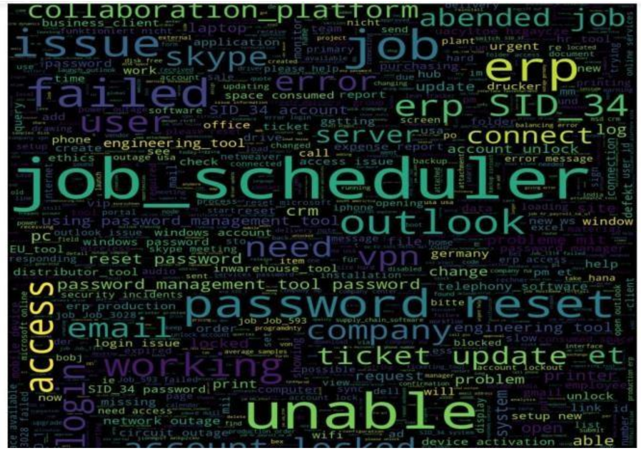


Analysis using Word Clouds

Descriptions WordCloud



- **Short Descriptions WordCloud**



WordCloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or relative importance within the dataset. Significant textual data points can be highlighted using a word cloud. WordClouds have been generated with All available words & top 50 words.
We have also inferred a few observations over the target class – Assignment groups with word clouds for the top 50 words from each group

In the security log cleanup, we have removed ip addresses, special characters, extra whitespace in the event data descriptions and the duplicate entries

This function is specific to the security/event logs present in the dataset which start with a specific pattern

Key Insights and Takeaways

74 Assignment groups found - Target classes

Group 0 is the majority class which accounts for ~47% of the data and the remaining groups are relatively much less frequent resulting in ***highly imbalanced data***.

Around 17% of the descriptions were found to be in ***Non-English languages***

Several Emails were found in the description

Some descriptions have entire security/event logs

Symbols & other non-ascii characters were detected in the description

Hyperlinks, URLs, Email Addresses, Telephone Numbers & other irrelevant information was found in the descriptions

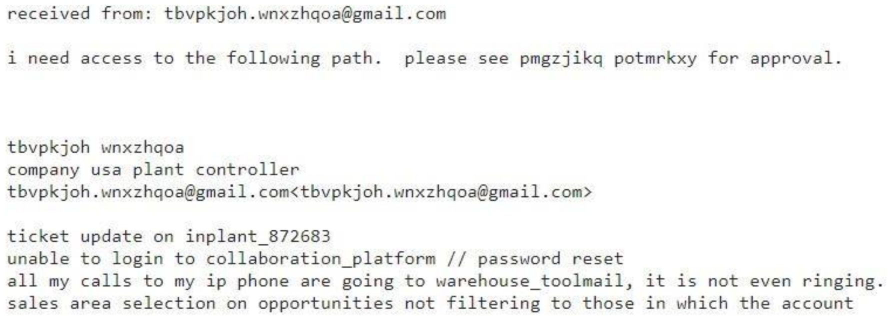
Blanks found either in the short description or description field

Few descriptions same as the short description

Contraction words found in the merged Description and expanded for ease of word modelling

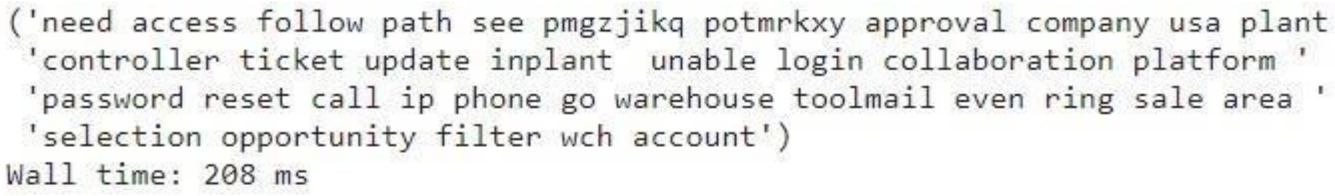
Final Pre-Processing Techniques applied

Output before Pre-Processing



```
received from: tbvpkjoh.wnxzhqoa@gmail.com

i need access to the following path.  please see pmgzjikq potmrkxy for approval.



tbvpkjoh wnxzhqoa
company usa plant controller
tbvpkjoh.wnxzhqoa@gmail.com<tbvpkjoh.wnxzhqoa@gmail.com>

ticket update on inplant_872683
unable to login to collaboration_platform // password reset
all my calls to my ip phone are going to warehouse_toolmail, it is not even ringing.
sales area selection on opportunities not filtering to those in which the account
```

Output after Pre-Processing



```
('need access follow path see pmgzjikq potmrkxy approval company usa plant
 'controller ticket update inplant  unable login collaboration platform '
 'password reset call ip phone go warehouse toolmail even ring sale area '
 'selection opportunity filter wch account')
Wall time: 208 ms
```

Below steps have been performed for initial pre-processing and clean-up of data in the preprocess_text function:

Fix text encoding using ftfy.fix_text — A lot of text in the data was being misinterpreted as some gibberish text

(æ‰“å¼€ outlook) when in reality they were Chinese characters (打开 outlook)

Parse email messages to retain only subject and body — Parse the mails to strip out headers, salutations, attachments etc., to retain only the relevant message.

Clean up emails, links, website links, telephone numbers — Strip out any of this unnecessary information using regex patterns.

Clean up anchor words like: 'Received from:', 'name:', 'hello', 'hello team', 'cid'... etc., — Strip out any of these filler words which add no information to the model

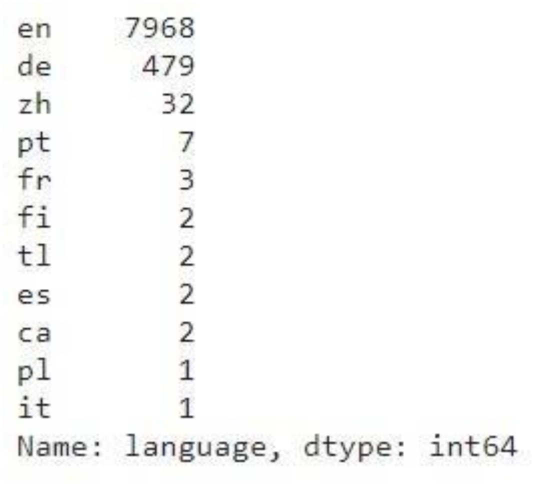
Clean up security logs — Clean the logs in the data by removing unnecessary information using regex patterns Clean HTML tags wherever they exist in the data
Clean Blank (/r /n) characters
Strip caller names in descriptions — Caller names were found to be present in the descriptions as well, these values were tokenized and stripped out if found in the descriptions
Translate/Normalize accented characters (á -> a)
Convert Unicode characters to Ascii
Expand contractions (they're -> they are)
Clean stopwords & a few custom stopwords were found by analyzing the text
Clean up extra whitespaces between words & Tokenize
Remove gibberish — A lot of gibberish was still found to be in the text, this was stripped out using regex patterns Remove extra punctuation
Changed the case sensitivity of words to lowercase

Language Translation

The objective is to detect presence of Non-English languages in the dataset and using translation technique to preserve insights from non english text data

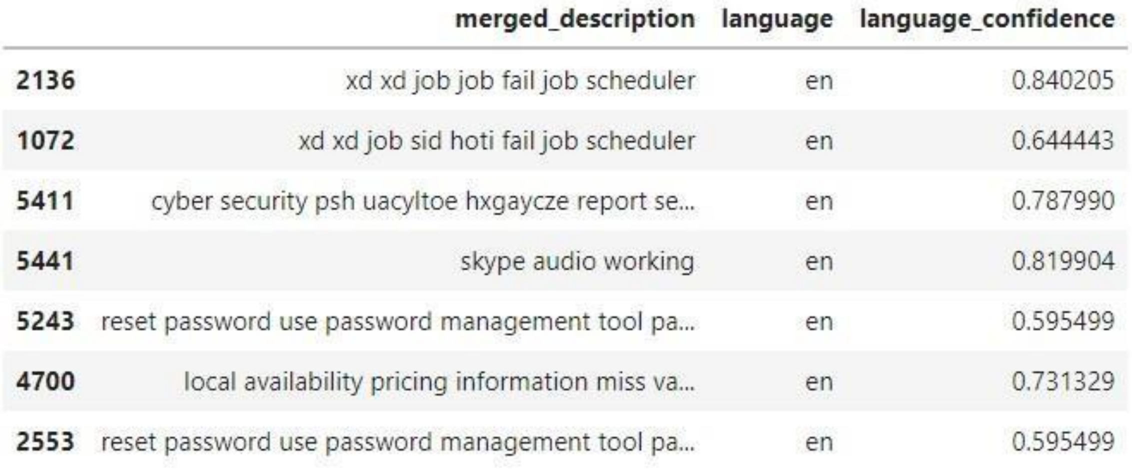
Detection of Non English Languages using fastText language identification model:

With the help of this pre-trained model, we were able to detect the presence of non english languages such as German, Mandarin, Portuguese, French etc.,



```
en    7968
de     479
zh      32
pt       7
fr       3
fi       2
tl       2
es       2
ca       2
pl       1
it       1
Name: language, dtype: int64
```

We also measured the language confidence metrics to consider correctness of language detection with language confidence > 0.6, and the descriptions with language confidence < 0.6 are considered as English



	merged_description	language	language_confidence
2136	xd xd job job fail job scheduler	en	0.840205
1072	xd xd job sid hoti fail job scheduler	en	0.644443
5411	cyber security psh uacyltoe hxgaycze report se...	en	0.787990
5441	skype audio working	en	0.819904
5243	reset password use password management tool pa...	en	0.595499
4700	local availability pricing information miss va...	en	0.731329
2553	reset password use password management tool pa...	en	0.595499

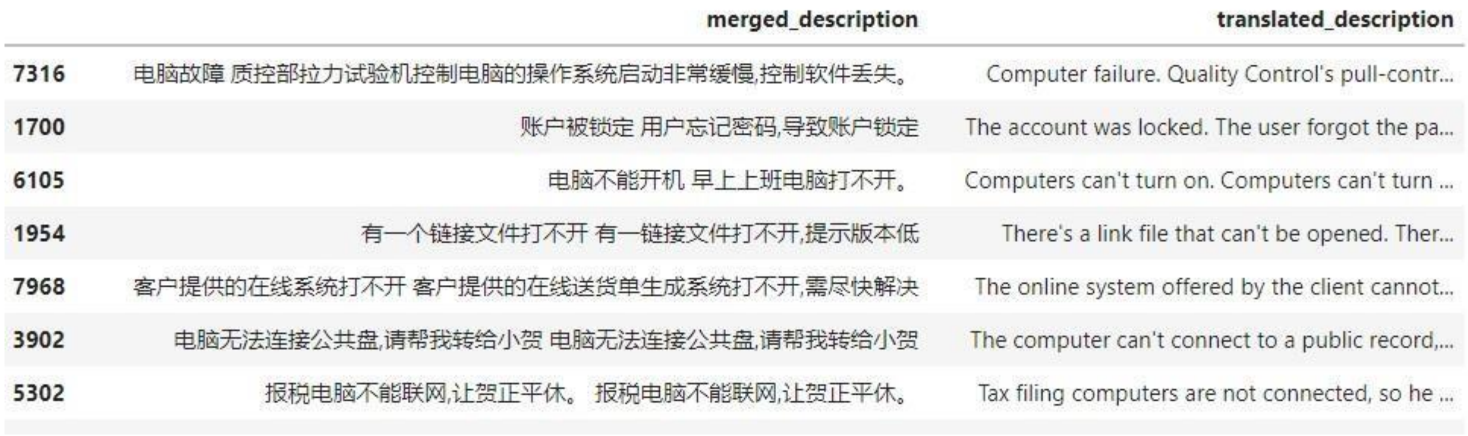
Translating Non English Languages:

Translation using txtai pipelines which use hugging-face language translation models as backend

The pipeline has logic to detect the input language, by loading the relevant model,it handles translating from source to # target language and return results.

The translation pipeline also has built-in logic to handle splitting large text blocks into smaller sections the models can handle

Sample Translations

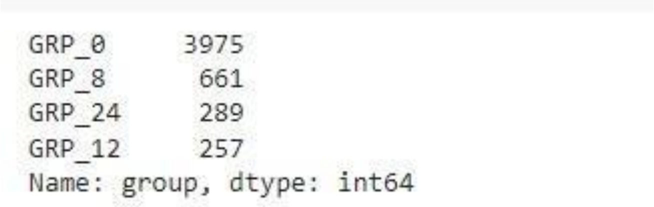


	merged_description	translated_description
7316	电脑故障 质控部拉力试验机控制电脑的操作系统启动非常缓慢,控制软件丢失。	Computer failure. Quality Control's pull-contr...
1700	账户被锁定 用户忘记密码,导致账户锁定	The account was locked. The user forgot the pa...
6105	电脑不能开机 早上上班电脑打不开。	Computers can't turn on. Computers can't turn ...
1954	有一个链接文件打不开 有一链接文件打不开,提示版本低	There's a link file that can't be opened. Ther...
7968	客户提供的在线系统打不开 客户提供的在线送货单生成系统打不开,需尽快解决	The online system offered by the client cannot...
3902	电脑无法连接公共盘,请帮我转给小贺 电脑无法连接公共盘,请帮我转给小贺	The computer can't connect to a public record,...
5302	报税电脑不能联网,让贺正平休。 报税电脑不能联网,让贺正平休。	Tax filing computers are not connected, so he ...

Target Class Balancing:

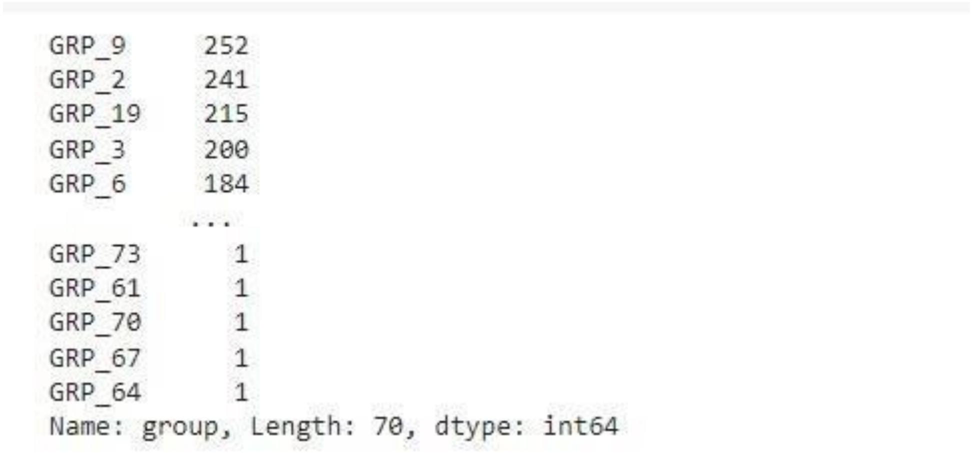
Transalated_description is our independent variable & *group* is our target class. As we see there are 74 group classes present & most of them are under-represented here. So, we have segregated groups with at least 3% share on dataset as major classes & remaining of them as minor classes.

Major classes:



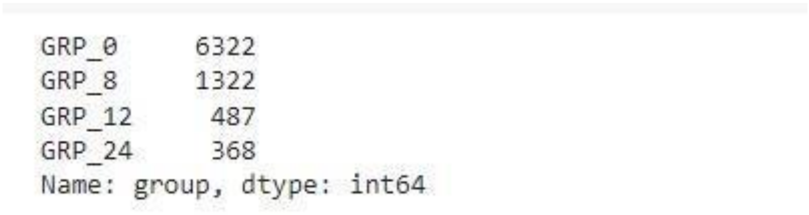
```
GRP_0     3975
GRP_8      661
GRP_24     289
GRP_12     257
Name: group, dtype: int64
```

Minor classes:



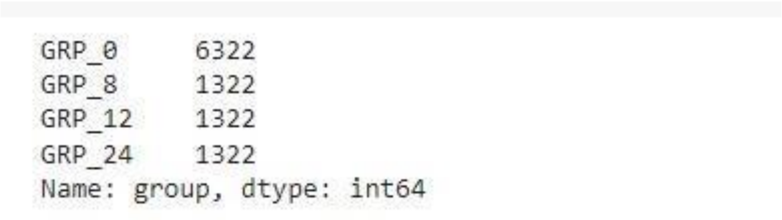
```
GRP_9     252
GRP_2     241
GRP_19    215
GRP_3     200
GRP_6     184
         ...
GRP_73      1
GRP_61      1
GRP_70      1
GRP_67      1
GRP_64      1
Name: group, Length: 70, dtype: int64
```

We’ve used *FuzzyWuzzy* library to find the best matching record in major classes set for each record in minor classes set & replaced that group value in minor classes set. Eventually we have both classes with all major group values & we merged them into as one & here is the resultant class distribution.



```
GRP_0     6322
GRP_8     1322
GRP_12     487
GRP_24     368
Name: group, dtype: int64
```

Now we have used traditional resample method for up sampling the lowest represented classes GRP_12, 24 to GRP_8.



```
GRP_0     6322
GRP_8     1322
GRP_12    1322
GRP_24    1322
Name: group, dtype: int64
```

We’ve split this dataset into train-test as 60-40% respectively.

ML Modelling:

Modelling for this problem has been presented into 3 sections. Each section contains Stacking as well.

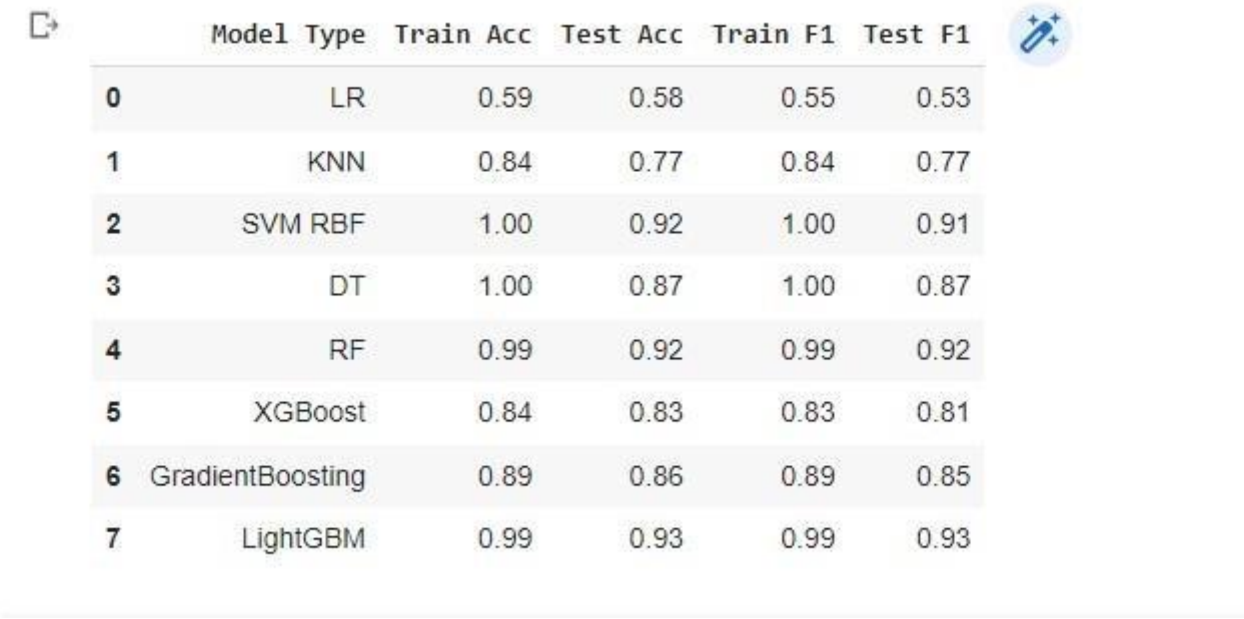
Section 1: All the base ML models with default parameters

Section 2: Cross-validation on all the base ML models

Section 3: All ML models with best parameters after hyper-tuning the base models

Section-1:

We have presented the F1 score along with Accuracy scores for better understanding the model performance.

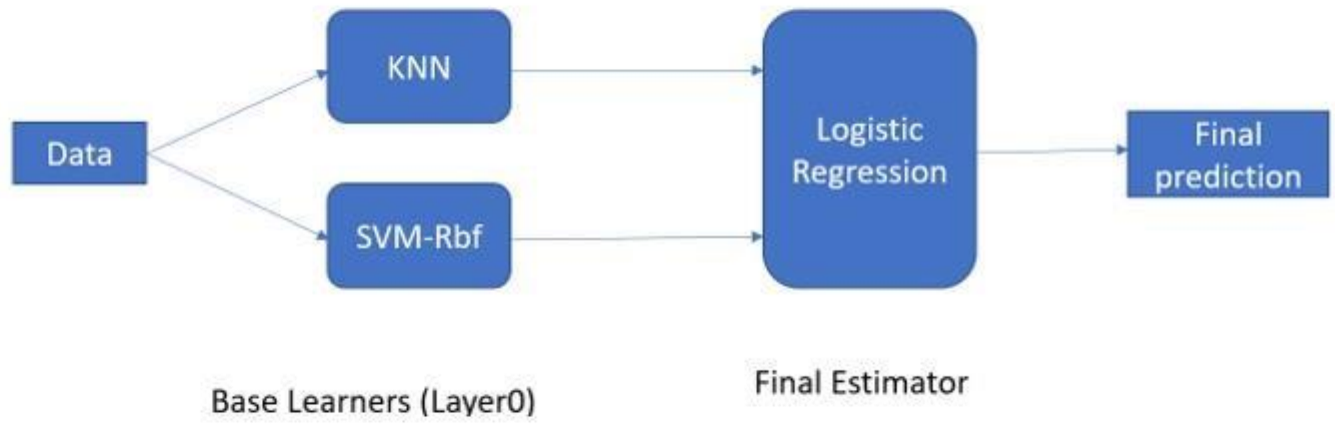


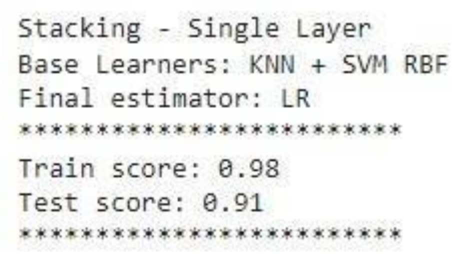
	Model Type	Train Acc	Test Acc	Train F1	Test F1
0	LR	0.59	0.58	0.55	0.53
1	KNN	0.84	0.77	0.84	0.77
2	SVM RBF	1.00	0.92	1.00	0.91
3	DT	1.00	0.87	1.00	0.87
4	RF	0.99	0.92	0.99	0.92
5	XGBoost	0.84	0.83	0.83	0.81
6	GradientBoosting	0.89	0.86	0.89	0.85
7	LightGBM	0.99	0.93	0.99	0.93

- Logistic Regression is under-performed as our data is linearly inseparable
- KNN, boosting methods performing decently well too
- Decision Trees, Random Forest methods with default parameters are over-fitting - Overall, we see the over-fitting problem & F1 score is little less than Accuracy

Stacking:

Single Layer:



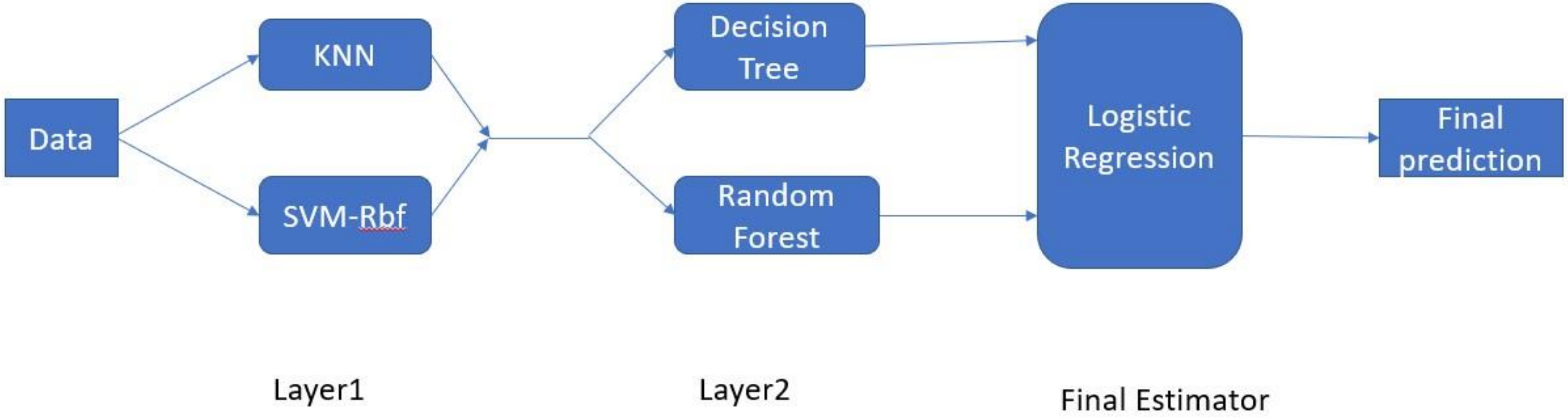


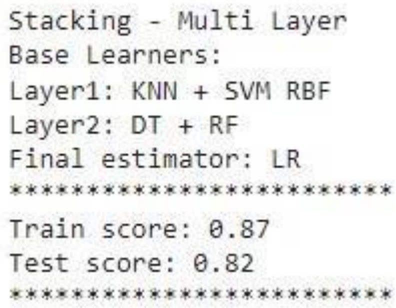
```
Stacking - Single Layer
Base Learners: KNN + SVM RBF
Final estimator: LR
*************************
Train score: 0.98
Test score: 0.91
*************************
```

As we see over-fitting problem persists even after Stacking.

Multi-layer:

We are combining the predictions from Layer1 and serving it as input to Layer2 here.

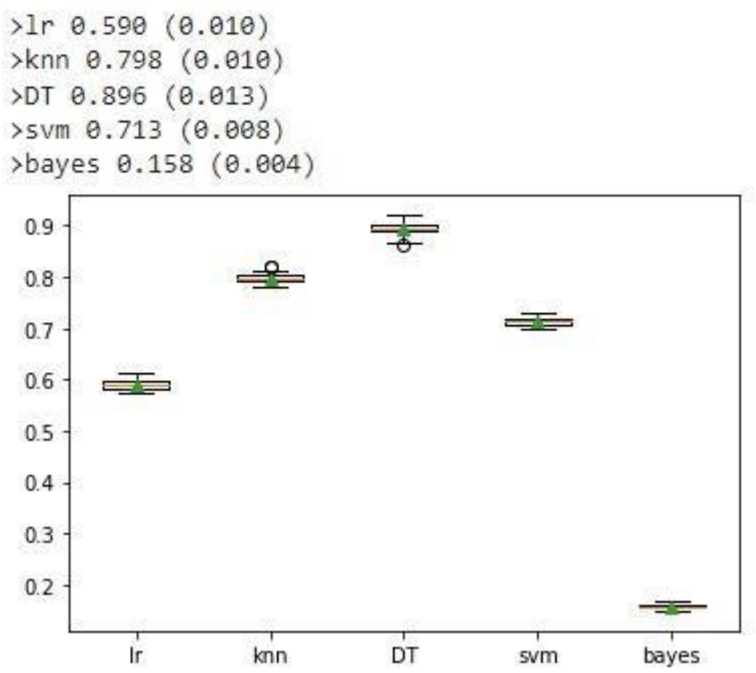




```
Stacking - Multi Layer
Base Learners:
Layer1: KNN + SVM RBF
Layer2: DT + RF
Final estimator: LR
*************************
Train score: 0.87
Test score: 0.82
*************************
```

Multi-layer did decently well. Let’s us do cross-validation on dataset too.

Cross-validation Methods:

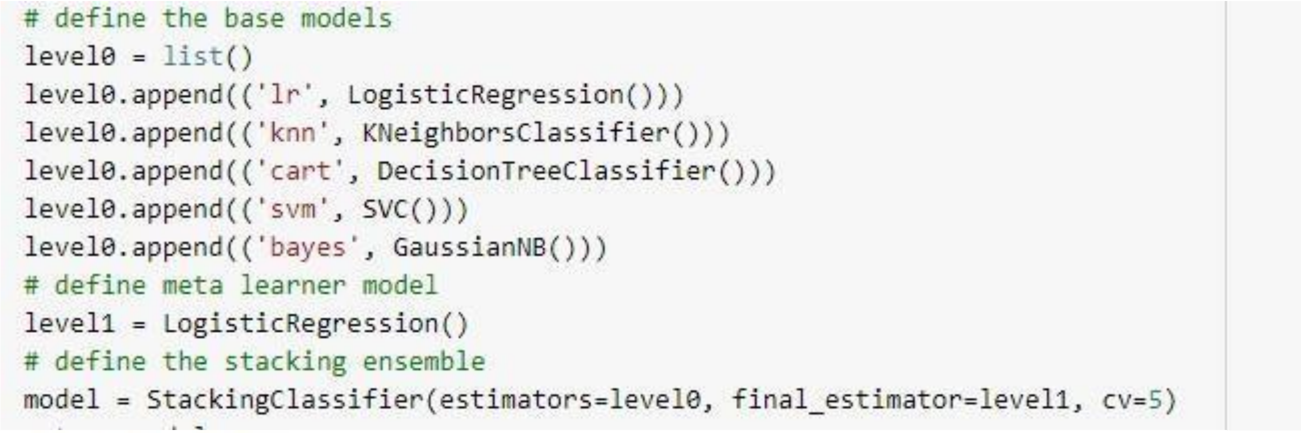


```
>lr 0.590 (0.010)
>knn 0.798 (0.010)
>DT 0.896 (0.013)
>svm 0.713 (0.008)
>bayes 0.158 (0.004)
```

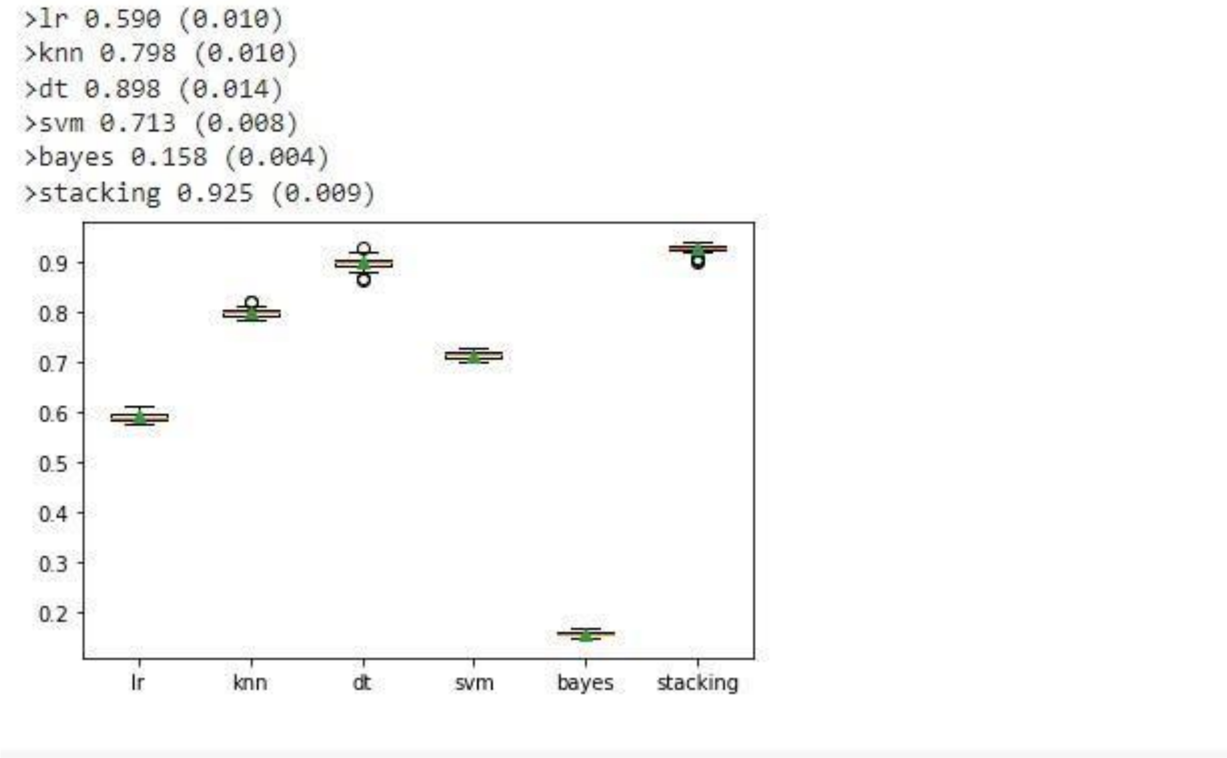
Decision Trees is performing well with least deviation.

CV with Stacking:

With slight change in base models & let’s build a stacking model here.



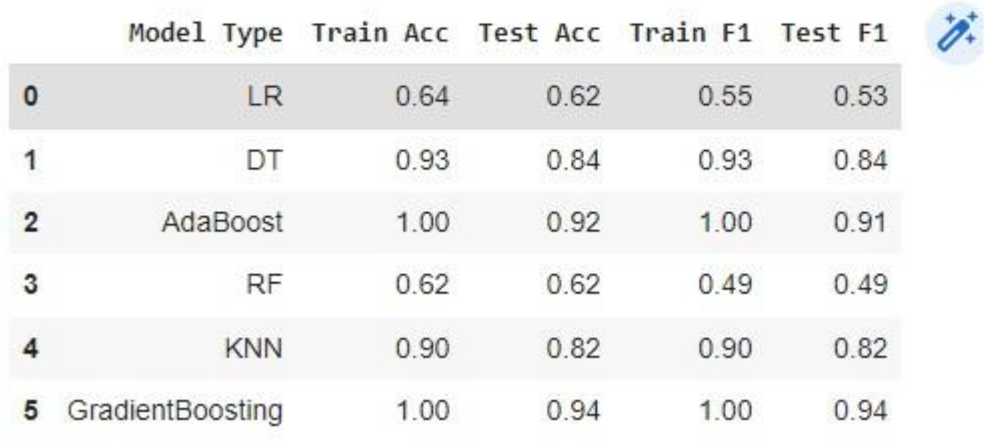
```
# define the base models
level0 = list()
level0.append(('lr', LogisticRegression()))
level0.append(('knn', KNeighborsClassifier()))
level0.append(('cart', DecisionTreeClassifier()))
level0.append(('svm', SVC()))
level0.append(('bayes', GaussianNB()))
# define meta learner model
level1 = LogisticRegression()
# define the stacking ensemble
model = StackingClassifier(estimators=level0, final_estimator=level1, cv=5)
```



```
>lr 0.590 (0.010)
>knn 0.798 (0.010)
>dt 0.898 (0.014)
>svm 0.713 (0.008)
>bayes 0.158 (0.004)
>stacking 0.925 (0.009)
```

Stacking is performing well with less deviation overall across all folds.

Hyper Tuning:



	Model Type	Train Acc	Test Acc	Train F1	Test F1
0	LR	0.64	0.62	0.55	0.53
1	DT	0.93	0.84	0.93	0.84
2	AdaBoost	1.00	0.92	1.00	0.91
3	RF	0.62	0.62	0.49	0.49
4	KNN	0.90	0.82	0.90	0.82
5	GradientBoosting	1.00	0.94	1.00	0.94

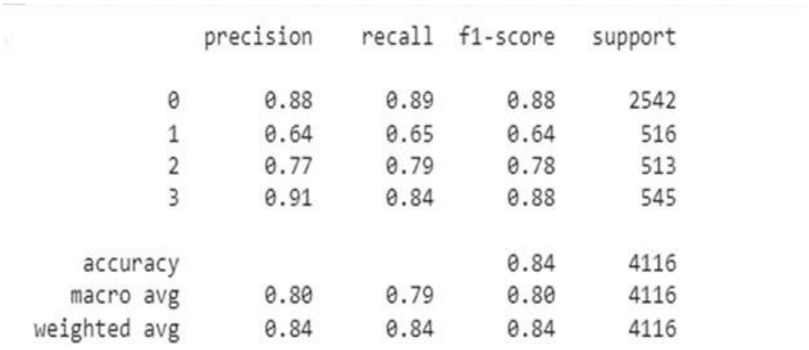
Logistic Regression isn’t improving even after hyper tuning.

With restricting the default depth & all, Random Forests and Logistic Regression are under-fitting. Let’s compare the classification reports of both Decision Tree model (our best) with Logistic Regression (underfitting) model.

Boosting models & KNN have been doing an exceptional job here.

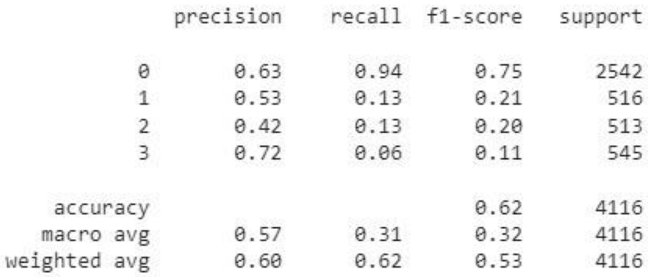
Let’s see how each class is being predicted by our model using classification reports.

Decision Tree classification report:



```
              precision    recall  f1-score   support

           0       0.88      0.89      0.88      2542
           1       0.64      0.65      0.64       516
           2       0.77      0.79      0.78       513
           3       0.91      0.84      0.88       545

    accuracy                           0.84      4116
   macro avg       0.80      0.79      0.80      4116
weighted avg       0.84      0.84      0.84      4116
```

Logistic Regression classification report:

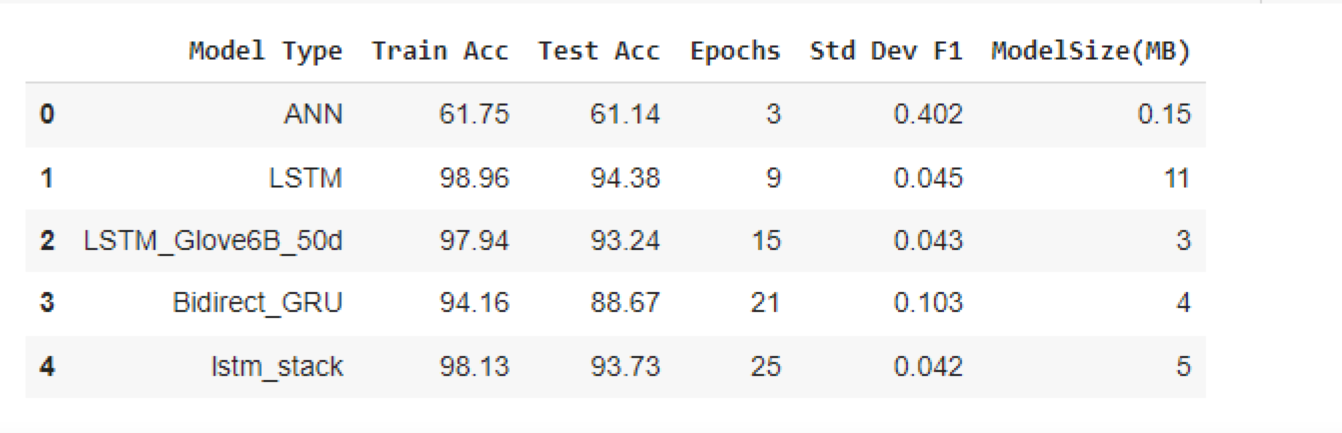


```
              precision    recall  f1-score   support

           0       0.63      0.94      0.75      2542
           1       0.53      0.13      0.21       516
           2       0.42      0.13      0.20       513
           3       0.72      0.06      0.11       545

    accuracy                           0.62      4116
   macro avg       0.57      0.31      0.32      4116
weighted avg       0.60      0.62      0.53      4116
```

Under-fitting models have terrible F1-score values for lower-represented groups such as 1,2,3. Our models are highly biased towards group 0, since it has ~50% data of whole. Even if we up sample the lower classes to as equal to group 0, we would be introducing high variance with synthetic data.

Deep Learning Models:

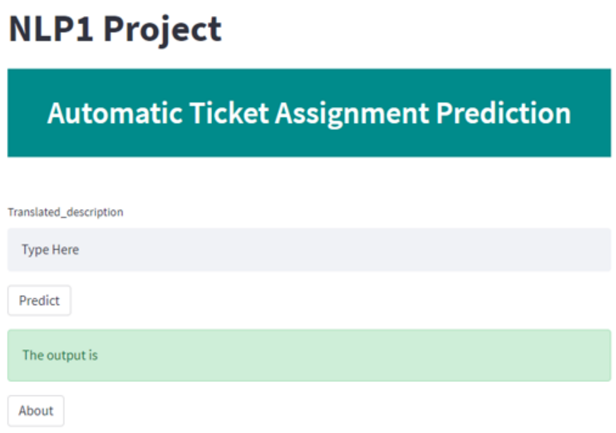
- Simple ANN with basic layers, is under-fitting model.
- LSTM is performing good with one LSTM layer and is occupying more physical space compared to all other models.
- LSTM single layer’s performance is much better with pre trained GloVe embeddings.
- Bi-directional GRU is good enough too, but not as much as LSTM level.
- With 2 LSTM layers, LSTM Stacking is also performing equally better.



	Model Type	Train Acc	Test Acc	Epochs	Std Dev F1	ModelSize(MB)
0	ANN	61.75	61.14	3	0.402	0.15
1	LSTM	98.96	94.38	9	0.045	11
2	LSTM_Glove6B_50d	97.94	93.24	15	0.043	3
3	Bidirect_GRU	94.16	88.67	21	0.103	4
4	lstm_stack	98.13	93.73	25	0.042	5

Deployment:

We have used **StreamLit** Library for project deployment. This creates a simple webapp which reads translated description as input and displays the class number as output.



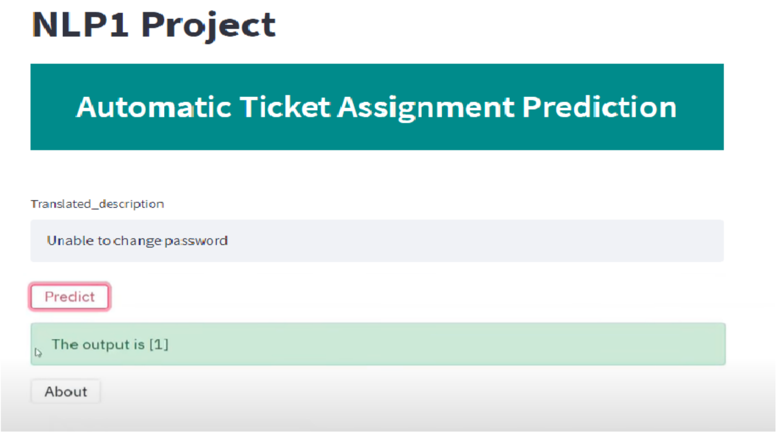
NLP1 Project

Automatic Ticket Assignment Prediction

Translated_description

Type Here

The output is



NLP1 Project

Automatic Ticket Assignment Prediction

Translated_description

Unable to change password

The output is [1]

Conclusion:

- Even after up sampling the data we’ve seen that the predictions are biased towards GRP_0, which has most samples.
- We also used hyperparameter tuning and stacked the different ML & DL models and then compared the results and picked the one with the best values.
- Had we balanced all the classes equally, we would’ve seen the high variance in our models anyway. So as part of balancing the variance & bias, we’ve tried the different weightage techniques to understand the relationship of minor classes as well.
- We’ve seen that most of traditional ML models are over-fitting. So, we tried the Stacking as well (single & multi-layer stacking) and seen that scores are lesser compared to base models. Stacking has brought out the stability we needed.
- We got below par performance with ANN model. It doesn’t pick up sequential data well.
- After padding with the fixed length, we’ve seen that the performance improved. Still ANN is underfitting though.
- Bidirectional GRU, LSTM is not outperforming the single layer LSTM. Because we have a very simpler dataset. Models like GRU, bi-direct LSTM would perform better on longer descriptions texts.
- Out of all DL & ML models, LSTM is doing much better job and even LSTM stacking too. Since LSTM gives attention to certain words & understands the relationship well.