Automatic IT Ticket Assignment

Final Report

Nargess Ghahremani

Introduction

Nowadays the vast majority of companies and businesses have an IT department to support the technological infrastructure they need. Depending on the nature of the business, the IT department has to provide support for the employees or business end users and there are costs - often significant - associated with any issues which either of them may face.
Therefore, responding to and solving the problems in a timely manner is critical. For an IT department to be able to resolve any issues efficiently, early detection of issues is paramount which is why most IT departments employ monitoring tools to identify any issues proactively, hopefully before they are noticed and reported by other users.

Once an issue is reported (by monitoring tools or users), the issue should be assigned to the right team/staff to be dealt with. Currently, in majority of cases, this is a manual process which if automated, could save significant time and money for the business as (a) it will assign the issue in the shortest time possible, (b) will save time from the support team, allowing them to help with resolution of issues and (c) if done well can be more accurate and as such saving resources which would have otherwise been wasted.

For this project, the current support process for the business is for the Level 1 and Level 2 users to assign the issues they cannot resolve to groups in Level 3 support. They need to review the SOP process, spending at least 15 minutes per ticket and at least 1 full time employee need to be dedicated to this. Their accuracy in assigning the ticket correctly is around 75%. Therefore, if by automating this process we achieve a higher accuracy than 75%, the business will:

1. Ensure issues are not waiting in queues before they can be assigned
2. Save having to dedicate 1 employee to ticket assignment
3. Achieve better results in assigning tickets

Exploratory Data Analysis

Initial Observations

Initial observation of the data reveals the following.

- The data consists of 8500 entries and 4 columns: short description, description, caller and group assignment.
- It consists of tickets raised by users (people) and by monitoring tools. It is not clear how frequent the data was collected and over which period of time.
- **Input Variables**: The data is not clean or consistent. It contains:
    - a few blank fields
    - symbols (both readable and illegible)
    - email formats
    - filled form format
    - references to image files
    - hyperlinks
    - URL's
    - references to caller names
    - non-English languages

  all of which need to be cleaned or treated.
- **Target Variable**: Tickets are assigned to 74 groups in a highly imbalanced manner. More than 50% of the tickets are assigned to Group 0 which by an initial assessment seems to cover generic account/password issues as well as being the fall back for anything that is not otherwise assigned.

Data Pre-processing Steps

The following steps were taken to clean and pre-process the data for better understanding it and to prepare it for feed it to models.

Data Cleaning

- Convert to string (some fields were treated as float)
- Fill NA values
- Detect the language of the 'Description' field and anylse the outcome (consider only German as correctly detected non-English language)
- Clean:
    - Symbols

    - Digits
    - Email formats
    - HTML, hyperlinks, image file references
    - Caller names
    - Additional white spaces
- Merge 'Short description' and 'Description' fields and drop the former

Translation:

Translated German entries to English.

Remove Stop Words

Remove English Stopwords after adding the word 'please'. It was discovered by visualising the word distributions that 'please' was frequently appearing for some assignment groups while it clearly does not add any context.

Lemmatisation

Lemmatisation was chosen over stemming for better readability and compatibility with pre-trained models.

Data Analysis Steps

Topic Modelling

Forming dictionaries and corpus, then using both Bag of Words and TFIDF for LDA modelling yield the following 10 topics.

Topics using BOW

Topic	Terms
1	Tool, connect, Microsoft, screen, engineering, unable, internet, language , summary, name
2	Erp, account, sid, ticket, lock, update, team, create, inc, computer
3	Outside, software, usa, unlock, may, Germany, day, average, supply, sam ple
4	Access, yes, na, site, deny, circuit, power, company, vendor, network
5	Password, reset, id, outlook, tool, request, change, error, file, service

6	Job, scheduler, abended, fail, sid, drive, portal, reporting, hana, folder
7	Group, window, ip, call, order, phone, number, sale, problem, show
8	Issue, hostname, user, event, company, work, device, email, server, use
9	Unable, error, pc, plant, login, add, printer, print, per, warn
10	Inside, tcp, asa, dst, src, acl, aug, exe, jul, internal

Topics using TFIDF

Topic	Terms
1	Request, skype, log, screen, time, bex, report, set, let, meeting
2	Update, software, client, team, inc, computer, create, inplant, supply, chain
3	Access, error, open, use, try, hi, ethic, hr, load, reporting
4	Yes, na, site, telephony, circuit, power, backup, outage, warehouse, status
5	Outlook, collaboration, platform, launch, wifi, location, excel, lean, freeze, project
6	Job, abended, scheduler, fail, es, hana, etime, portal, net, dp
7	Password, reset, erp, tool, ticket, work, new, email, user, need
8	Account, sid, lock, unable, issue, login, window, hostname, setup, connect
9	Add, address, switch, freundlichen, sw, two, one, display, ip, ap
10	Mit, printer, usa, disk, problem, print, connection, audio, total, today

Although both models had similar coherence scores, BOW has segregated the topics better in terms of the inter-topic distances.

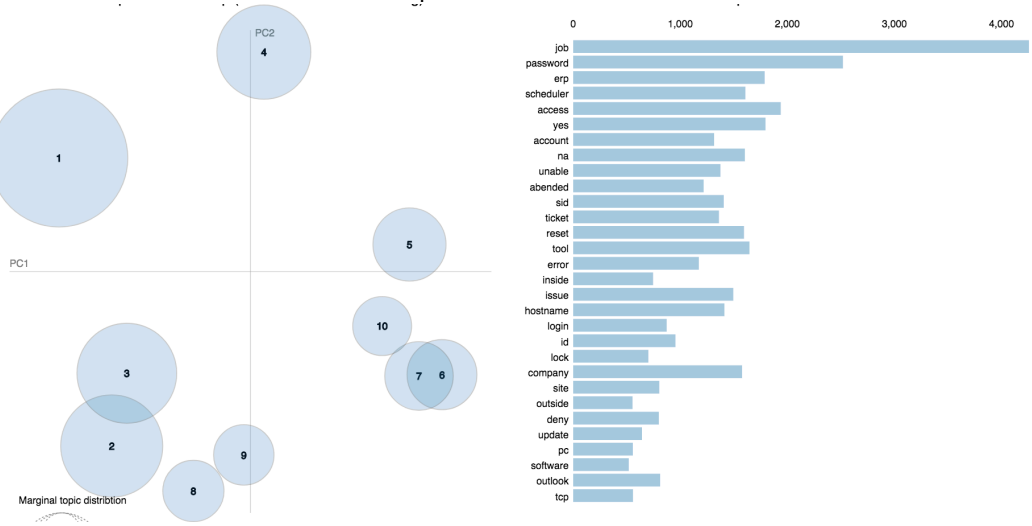


Figure 1- Topic Modelling using BOW

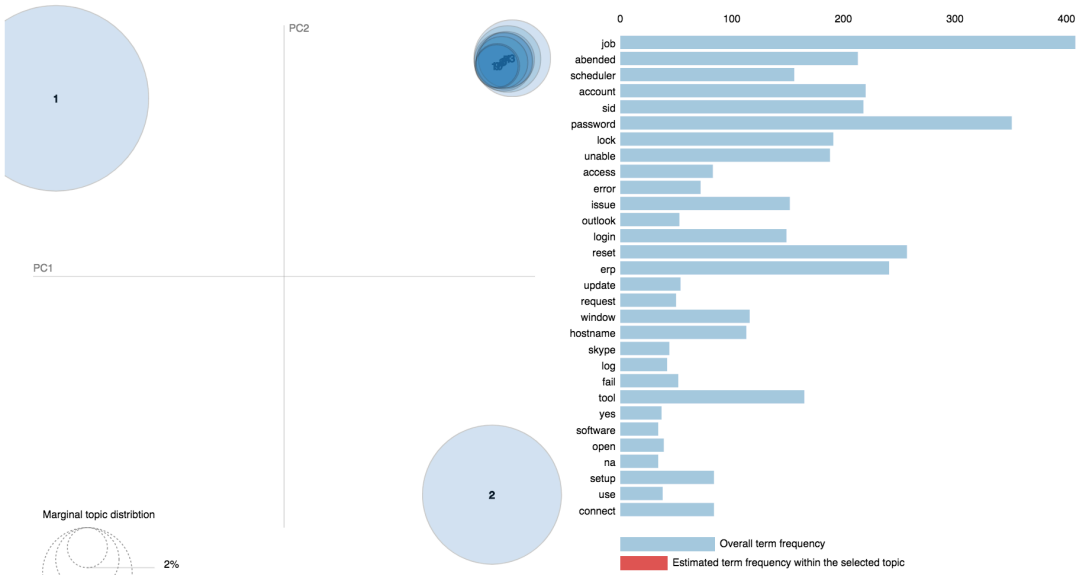


Figure 2 - Topic Modelling using TFIDF

Word distribution

Wordclouds of the whole corpus as well as those of the Assignment groups were created.

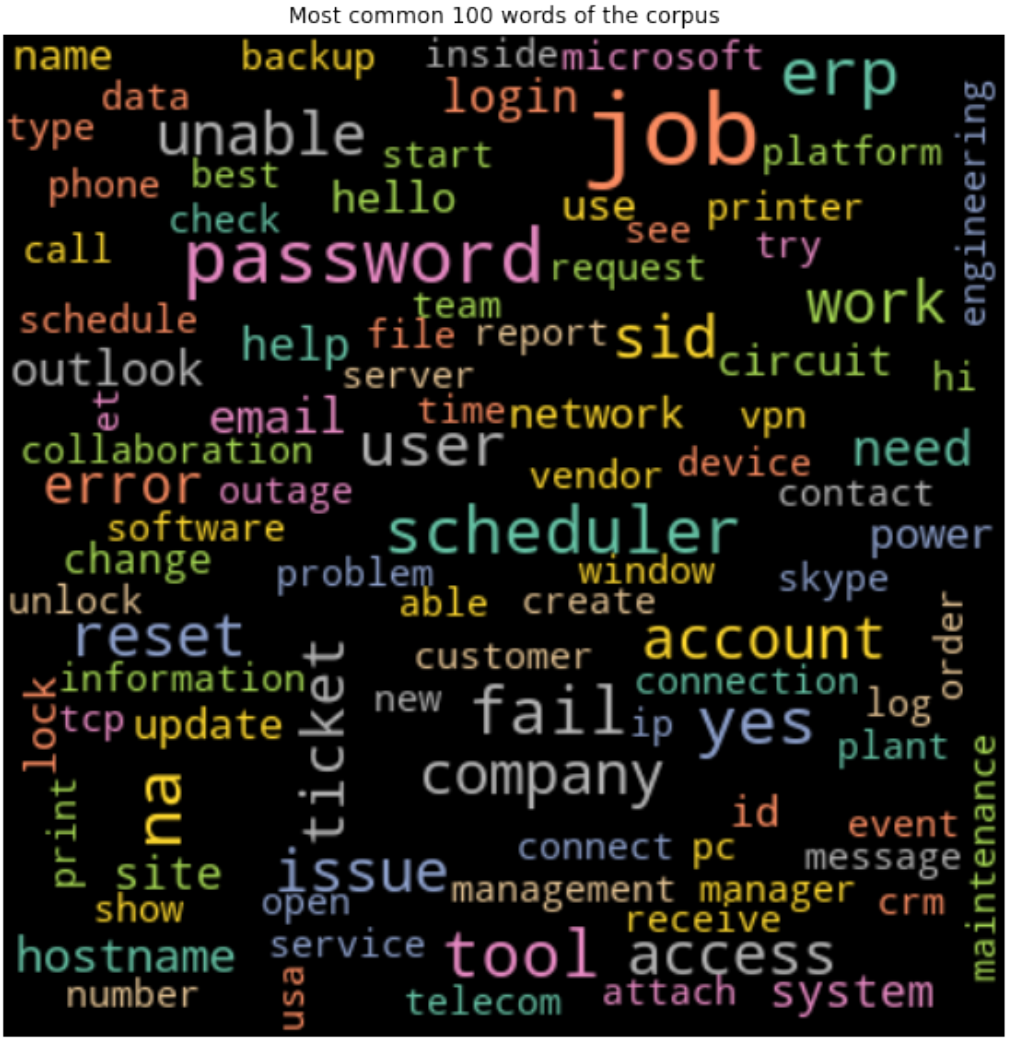
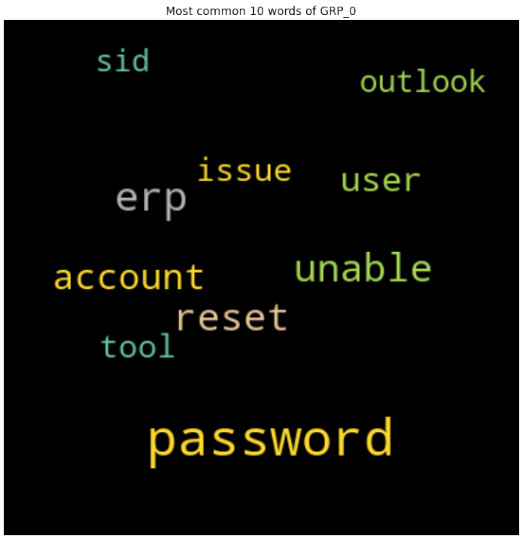
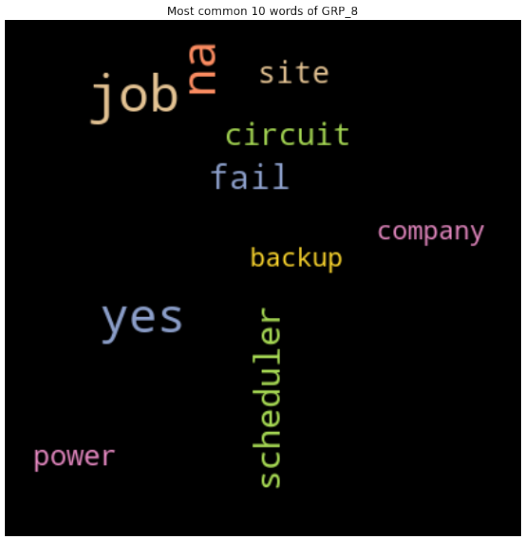


Figure 3 - Word Clouds





The 5 most assigned groups topic analysis is as follows.

Group	**Some of the most frequent terms**	**Topic**
Group 0	Password, unable, account, reset, outlook, user, erp	Account, password, outlook
Group 8	Job, scheduler, circuit, fail, site, backup, power, company	Power, network, monitoring tools
Group 24	Problem, setup, calculator, printer, tool, new	Printer and tools issues
Group 12	Hostname, server, access, drive, deny, available, disk, space	Server and hardware issues
Group 9	Scheduler, job, failed	Monitoring tools

'Language' and 'Caller' variables analysis

Initial observation of the tickets assigned per language suggests that the German tickets top 5 groups are Groups 0, 24, 33, 42 and 12, where the last three are not among the most frequent groups in general. Analysing the 'Caller' input variable suggests that there are certain callers whose tickets are assigned to a smaller number of groups. We therefore will consider including Lange and Caller as an input variable to see if they have an impact on our models' performance.

Treating imbalance in target variable

First and foremost, groups with less than 10 tickets assigned were merged together, leaving 50 groups in total.

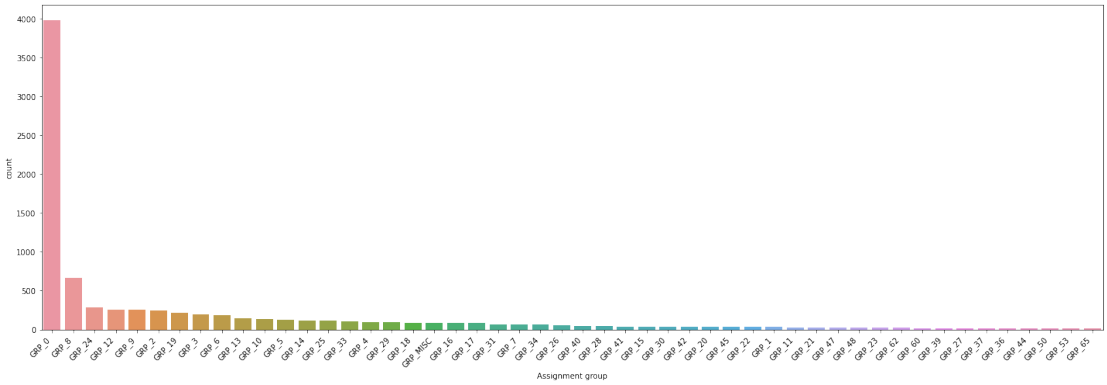


Figure 4 - Group Assignment Frequency

Then two approaches were considered in treating the imbalance in the target variable. First we analysed Group 0 (as the most frequent group) to see if any patterns can be recognised among this group. If clusters could be found in the group, we would then break Group 0 into further groups (either as standalone groups or to be merged with others based on similarity).
We used Doc2vec to convert ticket descriptions in Group 0 into vectors and fed the vectors to a KMeans model to form 5 clusters.

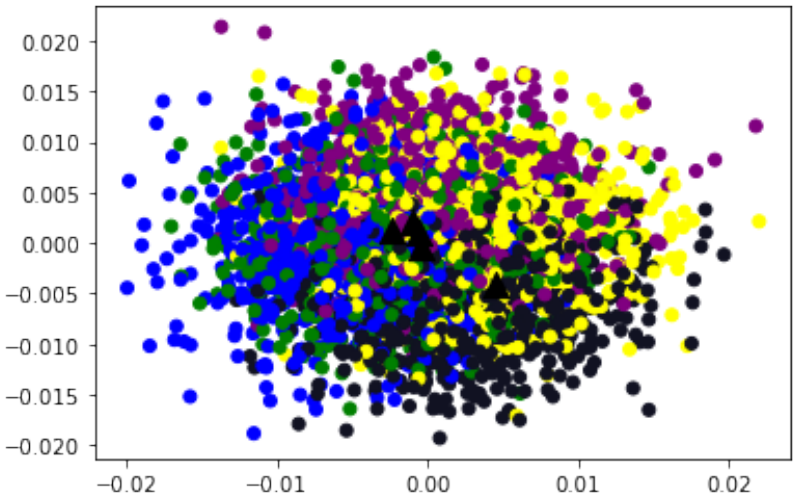


Figure 5 - Group 0 clusters

The result shows major overlap between the clusters, which suggests high similarity among tickets in Group 0. This suggests that the high frequency of tickets in Group 0 is not due to grouping separable topics to the same group, but because those issues are the most common ones. This fits with our topic

analysis that Group 0 is mostly about account/login and password issues. Furthermore, this group is sufficiently represented, supporting the reasoning behind our second approach of resampling.

Group 0 was down-sampled to the size of the second largest group (661 of Group 8) and the rest of the groups were up-sampled to the same.

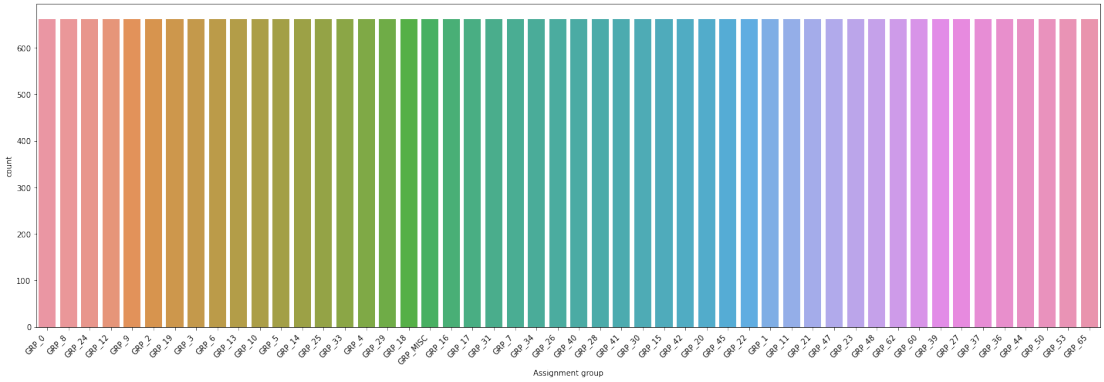


Figure 6 - Group Assignment Frequency after resampling

Modelling

The following lists several combinations of data, embeddings, and models which were tuned in order to achieve the best performance.

Embeddings - From Words to Numbers

As machine learning algorithms and neural networks cannot process text as input, we need to convert text to numbers. There are various approaches to this, from TFIDF to Word2Vec, Glove, etc.

For this part of the project we only used Word2Vec, but in the next steps (milestone 3) we will try the best performing model with Glove and FastText and compare the results.

1. Word2Vec

Word2Vec is itself a (shallow) neural network model which computes numerical vectors for words while learning the relation between them so that the semantic similarity between words corresponds to cosine similarity of the vectors.

2. Glove

Glove is based on matrix factorization techniques on the word-context matrix

And as such focuses on words co-occurrences over the whole corpus.
Its embeddings relate to the probabilities that two words appear together.

3. FastText

FastText improves on Word2Vec by also taking word parts into account. This enables training of embeddings on smaller datasets and generalization to unknown words.

Models

1. Bidirectional LSTM

Bidirectional LSTMs extend the classical LSTMs by using two LSTMs to process the sequence both as-is and in reverse (when both past and future of the sequence are available). By doing so they provide additional context to the network and can therefore improve the performance of the classical LSTMs.

2. Gated Recurrent Unit (GRU)

GRU's are in some way a simpler version of LSTM with a forget gate. They perform well for small and sparse datasets but LSTMs are strictly stronger than GRU's.

3. Deep Recurrent Network

Similar to adding multiple dense layers in a neural network, we could stack multiple layers of LSTMs on top of each other to achieve a more flexible model.

Hyper-tuning

Four different optimisers, namely Adam, SGD, RMSprop and Adagrad along with different learning rates were tested on the better perfoming models. Batch number and epochs were also played around with but are not included in this report in favour of keeping it short.

Data and Evaluation

We used Word2Vec embedding with bidirectional LSTM on 3 different versions of data:

a) Raw data (Description variable) where only the minority groups are merged into a Miscellaneous group

b) Resampled data (Description variable) where the data is resampled to be of the same size of 661.
c) Resampled data as above, with additional input variables of 'Language' and 'Caller'

The results show that resampled data achieves the best performance and adding additional variables does not have a significant impact on the performance. Therefore we continued to work with only the 'Description' input variable.

As we this is a multiclass classification problem, accuracy is not an adequate measure of performance, therefore, we rely on the F1-score as a better metric which incorporates both Recall and Precision metrics.

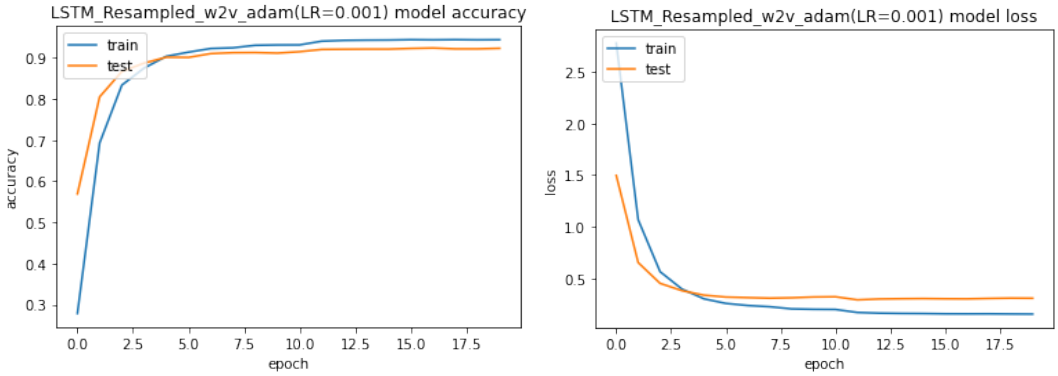
Results

Description	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Test Accuracy	Test F1 Score
LSTM_Raw_w2v_adam (LR=0.001)	0.575	1.866	0.598	1.643	0.544	0.514
LSTM_Resampled_w2v_adam (LR=0.001)	0.923	0.317	0.945	0.150	0.925	0.927
LSTM_Resampled_w2v_adam (LR=0.001) and additional variables	0.922	0.305	0.945	0.156	0.922	0.923
LSTM_Resampled_glv_adam (LR=0.001)	0.924	0.315	0.944	0.150	0.923	0.924
LSTM_Resampled_ftxt_adam (LR=0.001)	0.922	0.285	0.940	0.171	0.923	0.924
GRU_Resampled_w2v_adam (LR=0.001)	0.922	0.280	0.945	0.153	0.923	0.925
DRNN_Resampled_w2v_adam (LR=0.001)	0.920	0.366	0.946	0.149	0.921	0.923
LSTM_Resampled_w2v_sgd (LR=0.1)	0.883	0.368	0.883	0.363	0.880	0.879
LSTM_Resampled_w2v_rmsprop (LR=0.001)	0.922	0.325	0.944	0.156	0.922	0.924
LSTM_Resampled_w2v_adagrad (LR=0.1)	0.913	0.313	0.918	0.236	0.914	0.915

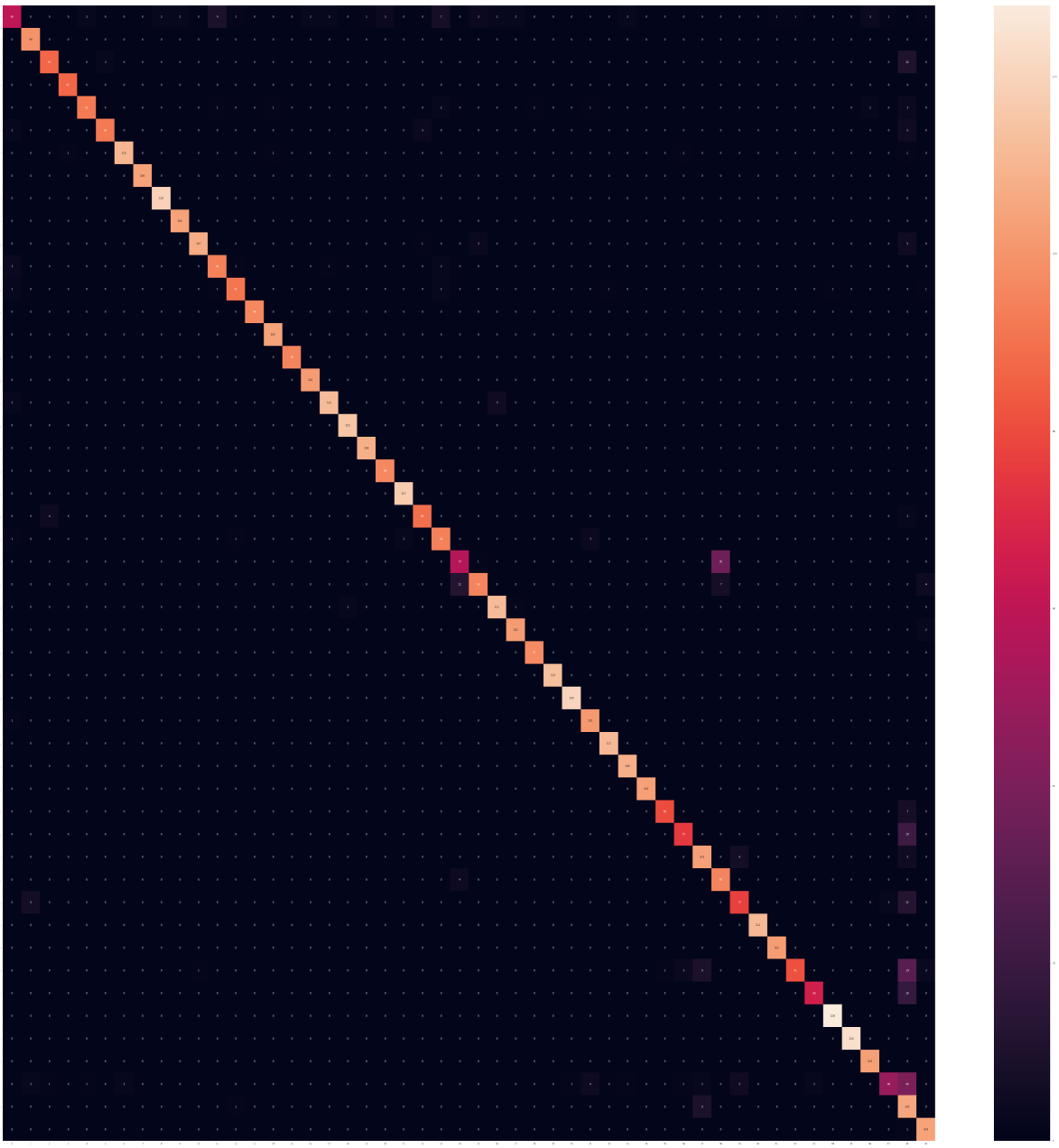
As the above results show, most models have very similar performance (except for LSTM on raw data and LSTM with SGD optimiser). As we have to choose one model, we pick the best performing one, which is the **bidirectional LSTM** using **Word2Vec** embedding and **Adam** optimiser. We train the model 5 times with

different splits of data to gain some confidence on the result. The outcome is as follows and is promising as the variation is miniscule.

```
Average Accuracy:      0.925
Min Accuracy:          0.924
Max Accuracy:          0.925
Variance of Accuracy:  4.778e-7
Average F1 Score:      0.926
Min F1 Score:          0.925
Max F1 Score:          0.927
Variance of F1 Score:  5.923e-7
```



Next we look into the confusion matrix for the best performing model to gain some insight into misclassified tickets.



Analysing the above confusion matrix, the vast majority of misclassifications belong to certain tickets (mostly from groups 5,6,8,10,45 and 60) which are being incorrectly assigned to Group 9 (label encoded to 48 – the column before last). Looking into the tickets for these groups, it becomes evident Group 9 has mostly tickets raised by monitoring tools (with the description "job fail scheduler" after text cleaning and preprocessing). These tickets simply do not have sufficient information that would make them distinguishable from other similar ones of the other groups.

Other patterns of misclassifications are smaller in number and seem to be related to the tickets which had many unrecognisable symbols and as such had minimal data after data cleaning. It would help to understand the nature of those tickets to avoid such misclassifications.

There are groups with 100% accuracy and others which have some misclassifications but in a random manner with no apparent patterns. For example Group 0 has 50+ misclassified tickets but they are scattered across other groups.

Business Recommendation

Our model has achieved a confident accuracy (and F1-score) of 92% which is considerably better than the human performance of 75%. This can further be improved by following some suggestions:

1. Minority groups that were merged as part of data preparation should either be grouped as one miscellaneous group, or more data is required to make them distinguishable.
2. Tickets generated by monitoring tools can be tweaked to include more information that would help with automatically assigning them to the right group. For example, some textual information can be added to the email to explain what section/tool/department the failed scheduled job relates to. Taking a step further, emails from monitoring tools can be set to be directed the right group in the first place following a rule based approach which will achieve a 100% performance on this category of tickets.
3. The multitude of tickets in Group 0 suggests that there a high number of tickets are related to resetting accounts. Revisiting and redesigning automated recovery approaches for various accounts can heavily reduce the number of tickets generated.
4. Better, more balanced and cleaner data will always help to improve the performance.

Conclusion

In this project, by testing and tuning combinations of data, word embeddings, and models we have confidently achieved an accuracy and F1 score of %92 which is an

improvement of 17% on the accuracy of the existing solution of assigning tickets manually. This unfolds to the following achievements.

1. **Cost Saving**: $70K per annum - assuming there are on average 50 tickets per day and that each misclassification costs $10, the automation can save $85 per day, i.e. over $30K annually. This is in addition to the salary of having at least one dedicated employee for assigning tickets manually, costing on average $40K per annum.
2. **Time Saving**: 2121 days - Tickets will be assigned immediately without requiring 15 minutes to be spent on each ticket, thus saving 570 days. As for misclassified tickets, assuming there are on average 50 tickets per day and that each misclassified ticket would be delayed by 0.5 day, a daily delay of 4.25 equal to annual delay of 1551 days will be avoided.
3. **Business Recommendation**: Steps to improve business processes were suggested above that can further improve the performance, thus saving more cost and time while helping the business with tackling some of the fundamental issues resulting in high number of tickets being generated.