INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Port request classification automation through NLP

Samuel António Beecher Martins

Master in Information Systems Management

Supervisor:
Doctor, Nuno Miguel de Figueiredo Garrido, Assistant Professor,
Iscte-IUL

August, 2024

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Port request classification automation through NLP

Samuel António Beecher Martins

Master in Information Systems Management

Supervisor:
Doctor, Nuno Miguel de Figueiredo Garrido, Assistant Professor, Iscte-IUL

August, 2024

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr Nuno Miguel de Figueiredo Garrido, for his unwavering support, guidance, and expertise throughout this research. Dr. Nuno Garrido has not only provided invaluable insights and constructive feedback for my dissertation, but he has also been instrumental in assisting me with other academical subjects. His dedication to my academic growth and willingness to help beyond the scope of this project have been truly remarkable. I am deeply appreciative of his mentorship and the profound impact he has had on my academic journey.

I would like to extend my sincere thanks to the members of the jury. I am grateful for their willingness to review my work and provide valuable feedback. I look forward to receiving your insight and suggestions which will undoubtedly enhance the quality of this dissertation. Your time and effort are greatly appreciated.

I would also like to thank Prof. Pedro Sebastião and the Instituto de Telecomunicações (IT), for their support which made it possible for me to present an article related to this dissertation.

To my family, and especially to my parents, Cecília and Fernando Martins who have been a driving force to complete this endeavour, and to my sister and her husband, Ester Martins and João Nunes for always lending a helping hand in accommodation and day to day motivation.

I am fortunate to have been surrounded by a group of inspiring and supportive colleagues and peers, from the academical field as from the labour field. Special thanks, your camaraderie and collaboration have made this journey enjoyable and intellectually stimulating. A special thanks to my work colleagues who, besides proving support, were subject to the inquiry and have taken their valuable time to answer it.

To friends, for motivating, supporting and to comprehend my absences I would like to extend my gratitude.

I would like to extend my heartfelt gratitude to my workplace, Administração do Porto de Sines e do Algarve, for their invaluable support in this research. I am deeply grateful l for the permission to use the organization's data and for allowing it to be the subject of my investigation. The access to this data and the insights gained have been crucial to the success of this dissertation. Your cooperation and support are greatly appreciated.

Resumo

O processo de criação de ambientes de trabalho sem papel, aliado à crescente interação dos utilizadores que interagem com aplicações de terceiros, tem conduzido à implementação de sistemas de *helpdesk*. No entanto, estes sistemas apresentam falhas na resolução de problemas, devido a dificuldades na pré-análise do pedido, ao elevado tempo de resposta, bem como a erros humanos ocasionais.

Este projeto descreve um protótipo capaz de realizar automaticamente a rotina de classificação de pedidos de uma ferramenta de *helpdesk* em ambiente de gestão de Portos. O protótipo é proposto como uma implementação válida desta *framework* para verificar a sua viabilidade para este sector.

Vários modelos de aprendizagem foram utilizados e testados durante este trabalho, tais como: SVM; *Decision Tree*; *Random Forest*; LSTM; BERT e um modelo hierárquico SVM. Para verificar a eficiência desses modelos, utilizamos o F1-Score como métrica de validação. Obtivemos F1-Scores de 95,42%; 93,47% e 77,23% ao classificar a categoria, o grupo e o subgrupo do pedido, respetivamente, utilizando um modelo hierárquico onde o grupo foi classificado usando um modelo SVM e onde a categoria e subcategoria foram classificadas com um modelo BERT.

Por fim, foi lançado um breve questionário quantitativo e longitudinal para determinar a forma como a equipa de *helpdesk* encara o sistema de distribuição dos pedidos de *helpdesk*. Este foi considerado crucial pela equipa, com uma opinião consistentemente elevada sobre a sua importância atual no fluxo de trabalho, e com uma confiança geral nos esforços e na eficácia de uma solução para automatizar esta atividade.

Palavras-Chave: Helpdesk, Classificação de Pedidos, Administração Portuária, Machine Learning.

Abstract

The process of creating paper-free work environments, allied with the increasing interaction of users who interact with third party applications has led to the implementation of helpdesk systems. These are usually associated with specific departments, in the entities promoting information services to promote support and end-user satisfaction. However, these systems have flaws in problem solving due to difficulties in pre-analysis in the request, high response time and occasional human error.

This project describes a prototype to automatically carry out the routine classification of requests from a help desk tool in the Port management environment. The proposed prototype is presented as a viable implementation for this sector.

Several simulation models were employed and extensively tested during this work, such as SVM, Decision Tree, Random Forest, LSTM, BERT and a SVM hierarchical model. To verify the efficiency of these models we used F1-Score as a validation metric. We obtained F1-Scores of 95.42%, 93.47% and 77.23% when classifying the request's category, group and subgroup respectively, using a hierarchical model where the group was classified using a SVM model and where the category and subcategory were classified using BERT model.

Lastly, a brief quantitative and longitudinal questionnaire was launched to determine the way that the helpdesk team views helpdesk request distribution. The request classification was considered crucial by the team, which had a consistently high opinion of its current importance in the workflow. They also demonstrated considerable confidence in the potential effectiveness of such a solution in automating this activity.

Keywords: Helpdesk, Request Classification, Port Administration, Machine Learning.

Index

Acknowledgements ..... **i**
Resumo ..... **ii**
Abstract ..... **iii**
Index ..... **iv**
Tables Index ..... **vi**
Figures Index ..... **vi**
List of abbreviations ..... **vii**
Chapter 1 – Introduction **1**
1.1. Topic context 1
1.2. Motivation and topic relevance 2
1.3. Questions and research goals 3
1.4. Methodologic approach 3
1.5. Structure and organization of dissertation 4
Chapter 2 – Literature review **6**
2.1. Help desk request attribution 6
2.2. Request Classification 6
2.3. NLP methodologies applied to classification use cases 6
2.4. Implementation model limitations for this use case 8
Chapter 3 – Help desk classification **9**
3.1. Manual Classification 9
3.2. Vectorization methods 9
3.2.1 Bag of Words 10
3.2.1 Term Frequency – Inverse Document Frequency (TF-IDF) 10
3.3. ML models 10
3.3.1 Support Vector Machines 11
3.3.2 Naive Bayes 11
3.3.3 Logistic Regression 11
3.3.4 Decision Tree 11
3.3.5 Random Forest 12
3.3.6 Long short-term memory 12
3.3.7 BERT 13
3.4. GMP prototype 13
Chapter 4 – Dataset **15**
4.1. Data Extraction 15
4.2. Data Preparation for Subject 18

4.3. Data Preparation for Request Body 20
Chapter 5 – GMP Development 24
5.1. Request subject Category and Group classification 24
5.1.1. Non-hierarchical models 24
5.1.2. Hierarchical models 25
5.2. Request's Body Subcategory, Category and Group classification 26
5.2.1. Non-hierarchical models 26
5.2.2. Hierarchical models 27
Chapter 6 – Simulation Results 29
6.1. Request subject Category and Group classification 29
6.1.1. Non-hierarchical models 29
6.1.2. Hierarchical models 32
6.2. Request's Body Text for Subcategory, Category and Group classification 34
6.2.1. Non-hierarchical models 34
6.2.2. Vectorization experiments with non-hierarchical models 40
6.2.3. Hierarchical models 42
6.3. General conclusions 45
Chapter 7 – Questionnaire 47
Chapter 8 – Conclusion 52
References 54
Appendix A 57
Appendix B 58
Appendix C 59
Appendix D 60
Appendix E 61
Appendix F 62

Tables Index

Table 1 – Example of maritime var-char sequences..... 18
Table 2 – Group and Category request distribution 20
Table 3 – Subcategory distribution by Category 22
Table 4 – Group and Category Corelation..... 26
Table 5 – Email Subject non-hierarchical model results for Group classification. 30
Table 6 – Email Subject non-hierarchical model results for Category classification. ... 31
Table 7 – Email Subject hierarchical model results for Category classification. 34
Table 8 – Email Body text non-hierarchical model results for Group classification. 35
Table 9 – Email Body text non-hierarchical model results for Category classification. 37
Table 10 – Email Body text non-hierarchical model results for Subcategory classification. 37
Table 11 – Email Body text non-hierarchical models summary results. 39
Table 12 – Email Body text vectorization model results for Group classification..... 41
Table 13 – Email Body text vectorization experiments with SVM model. 42
Table 14 – Email Body text model results for hierarchical classification. 44
Table 15 – Questionnaire average response results. 49

Figures Index

Figure 1 – Manual distribution as is vs suggested distribution. 2
Figure 2 – GMP workflow 14
Figure 3 – Distribution of Word Counts in Email Subject 16
Figure 4 – Distribution of word counts in Email Body Text. 17
Figure 5 – Email subject SVM model for Group Classification non-hierarchical, confusion matrix. 30
Figure 6 – Email subject SVM model for Category Classification non-hierarchical, confusion matrix. 32
Figure 7 – Email subject SVM model for Category Classification as Hierarchical, confusion matrix. 33
Figure 8 – Email body text classification SVM model for Group Classification non-hierarchical, with base vector, confusion matrix. 35
Figure 9 – Email body text SVM model for Category Classification non-hierarchical, confusion matrix. 36
Figure 10 – Email body text SVM model for Subcategory Classification non-hierarchical, confusion matrix. 38
Figure 11 – Email body text SVM model for Group Classification non-hierarchical, with complete vector, confusion matrix. 41
Figure 12 – Email body text SVM model for Category Classification as Hierarchical, confusion matrix. 43

List of abbreviations

JUL – Janela Única Logística

JUP – Janela Única Portuária

PCS – Port Community System

ML – Machine Learning

NLP – Natural Language Processing

SVM – Support Vector Machines

BOW – Bag of Words

BERT – Bidirectional Encoder Representations from Transformers

RNN – Recurrent Neural Network

LSTM – Long Short-Term Memory

TF-IDF – Term Frequency – Inverse Document Frequency

ML – Machine Learning

DF – Data Frame

Chapter 1 – Introduction

1.1. Topic context

In the 21st century, the Portuguese port sector has been consistently and rapidly digitizing. Over two decades, two generations of Information Systems have already been implemented and they are currently in the transition to the third generation. This digital evolution has been characterised by the increased complexity of these systems.

In the first generation of Port Systems, only the direct actors involved in a ship's stay in port were included as direct participants in these systems. In the second generation, with a Port Community System (PCS), there was a great focus on communication with external entities, as well. This led to the inclusion of the railway module and the module for managing the stay of containers in the various Iberian logistics warehouses. The process of moving from a Port Community System to a National Single Window (NSW) system led to the third generation of web applications. These aimed to consolidate the passage of the Information systems to the NSW concept, as well as improve the PCS of each Portuguese Port Administration. These efforts would allow for the centralization of the various support applications of each port and add the concept of national layer and will further the dematerialization process [1].

These applications have dematerialized many of the processes that previously were executed directly by human intervention, and this forced many of the operators to embrace the digital sector. As a result of this transformation, many of the actors are now users of JUL application, and as such, they have been encountering some constraints and/or issues in their day-to-day operations while using the port applications. Hence, the Port Administrations have developed processes to support Users so that the constraints can be overcome while maintaining User satisfaction and confidence in these applications. This development has culminated in the creation of help desk departments, and these departments are the front-end contact with the user. They identify the constraints and proceed to their resolution, satisfying requests and questions, as well as delegating situations outside of their scope to the relevant department, when the situation so requires [2].

In this context Users contact the Help Desk Department of the Port Administration when they encounter the need for support via email and/or phone call. If they only contact via email this request creates a formal request for support. Once a formal request is created, it is categorized and manually assigned to a Help Desk team member, as shown on the left side of Fig.1.

This study proposes the creation of a prototype to perform the distribution, and categorization of help desk requests, in an automated manner, to reduce the costs in human resources associated with this process and optimize the process of pre-analysis of the formal requests received. It also features the development of a prototype that will allow for the collection and analysis of the results obtained and compare these to existing metrics of quality. Thus, transforming the process as is to the suggested distribution shown on the right side of Fig.1.

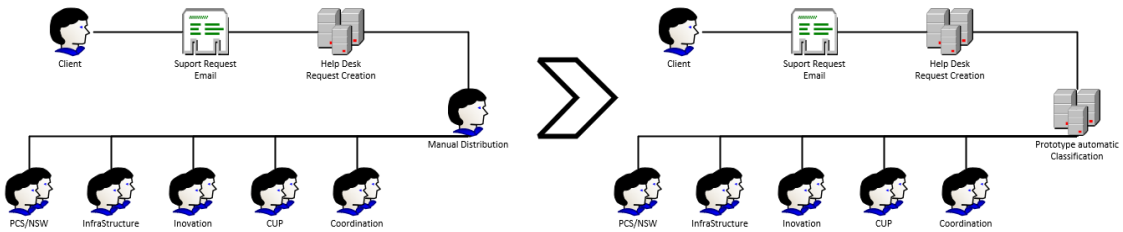

Figure 1 – Manual distribution as is vs suggested distribution.

1.2. Motivation and topic relevance

As explored by [3], the classification and passing on of requests is an area where users tend to have constraints. Moreover, as verified by [4], the distribution of these requests is relevant since an incorrect assignment introduces delays in the solution of the request. The constraints stem from the complexity of the portfolio of applications and services provided by the organisation, as well as the various work areas to which the request must be forwarded. Allied to these constraints, the lack of clarity in the information provided by the client creates further constraints in this pre-analysis of the request [5].

This study will create a model that allows for the classification of requests received in email format, in order to determine whether the automatic classification of these requests leads to an improvement in distribution times. It will also question whether this is an interesting technology for IT help desk teams in the port administrative sector at a national level. Finally, it proposes an implementation for this model as a functional prototype.

1.3. Questions and research goals

This dissertation intends to look at the following questions:

- How precise can an NLP model be while classifying a maritime Port help desk request?
- Which NLP model, from the options analysed, obtains the best results regarding the available dataset?
- Can the subject of a request alone suffice to obtain an accurate prediction or is the full treatment of the body required in order to obtain the best results?
- What is the level of acceptance of the members of the help desk team regarding this kind of technology and how is their perception of the technology affected when confronted with the pilot results?

1.4. Methodologic approach

The development of a prototype to automatically distribute and categorize help desk requests is aimed to reduce the human resources costs of this activity, to streamline the pre-analysis of incoming requests, and to gather valuable insights throughout the prototype's development process regarding this process.

The prototype will be a complementary tool to the existing help desk application, which, by pre-analysing the subject of the request and the body text classifies the requests. The development of this model will involve two versions: one focused on pre-analysing the subject of the request and another dedicated to pre-analysing the body text of the request:

- 1st Model, in a controlled environment the group and category of the request, will be determined automatically using the subject of the request.
- 2nd Model, in a controlled environment the group, category and subcategory of the request will be automatically determined, using the body of the request.

The data set is made up of all the requests that were created by the “Administração do Porto de Sines e Algarve” help desk system during 2022. They were mainly written in Portuguese. In total there were 19897 requests. We imported these 19897 requests to form the Main Data Frame (DF). The evaluation was to be carried out mainly by comparing

the F1-Score of the models presented above. Due to the unbalanced nature of this dataset other metrics such as Accuracy and Recall were also made available for analysis.

To calculate the F1-Score, Accuracy and Recall metrics, 20% of the requests present in the Main DF were used in a test DF to test the models. The remaining 80% of the requests present in the Main DF will be used to train the models.

To analyse the help desk team's level of acceptance and confidence in the prototype by, a questionnaire was drawn up and distributed to the management and help desk user groups. The same questionnaire was launched at each stage of the prototype, to allow for the monitoring of these actors' perceptions of the prototype. Each launch of the questionnaire included an update on the status of progress in implementing the prototype.

1.5. Structure and organization of dissertation

This dissertation is organized as follows:

- In chapter 2 we analyze some of the previous work that was executed in other fields in the implementation of NLP tools and the theorical aspects of these models and methods, with a special focus on unbalanced data sets.
- In chapter 3 we perform a brief overview of how the current manual classification is executed and how the automatization will act to better contextualize the reader with the current pain points and how this architecture will act.
- In chapter 4 we review the available data set for this dissertation, its characteristics and challenges, and how we have approached it, regarding pre-processing and mainstreaming the available labels.
- In chapter 5 we disclosed which NLP models were used and how they were configured for the different sources and origin of text, body versus subject.
- In chapter 6 we will review the results obtained by these models, discussing what the main pain points were in each in order to better understand the advantages or disadvantages of each model for the available data set.
- In chapter 7 we will cover the questionnaire, and the objective of the questions executed, then we will share the results obtained and will analyze the results obtained in each questionnaire session and look at what the evolution between each season reveals.

- In chapter 8 we will cover the conclusions that were reached with this study, comment on the limitations of this study and suggest future works that can be executed as a sequence of this study.

Chapter 2 – Literature review

This section is divided into three subsections, the first of which reviews the literature on other studies with similar characteristics, where the solutions and results have shown interesting results. It will also address some of the constraints detected by these studies. The second subsection will delve into the subject of Natural Language Processing (NLP), where we will also look at vectorisation and text pre-processing tasks. Finally, we will identify the models most frequently used in solutions with this theme.

2.1. Help desk request attribution

As the range of IT products evolves, users are often faced with constraints when using these products, whether due to difficulties in use, errors inherent in the product, poor application of the product, and other difficulties. In the course of these situations, some users turn to the IT product's support platforms in order to clarify their difficulty and, as mentioned by the authors [6] and [7], satisfaction with the service in this case will depend on the speed with which the request is answered, as well as its assertiveness. The result of this response will affect the user's perception of the product.

2.2. Request Classification

With the increased use of Customer Relationship Management (CRM) platforms, the implementation of Ticket Management Systems (TMS) is a valuable technical solution and tool both in the distribution, prioritization, and management of requests from customers [8].

In this way the classification of requests are intrinsically linked to the assignment of priority as well as to the service area of the entity providing the service. Depending on the project, if this is not used on a regular basis, the response time to the request as well as access to information about it is complicated, as is the case of the article [9]. On the other hand, as explored by [10], even when the organisation classifies requests consistently well, and follows good practices in this regard, the time that it takes to train operator's is a constraint. The specificity of some requests was also found to lead to the wrong classification by human operators at times.

2.3. NLP methodologies applied to classification use cases

Because the CRM system is the base for the classification process, many authors have tackled problems arising from this in different manners. For instant the author [11]

receives the customer information through the help desk application itself, being that the customer is enrolled in it. In this context an artefact was developed that applied vectorization to the term frequency–inverse document frequency (TF-IDF) model and subsequently to a Support Vector Machine (SVM) model for the classification of help desk requests for the German Jordanian University. In this case an accuracy rate of 83% was achieved when using body, subject and comments of the requests for classification.

The system developed in [12] performs classification by considering the body of the requests with a strong emphasis on comparison the classical SVM and Naive Bayes models. For this Corpora the Bagging-SVM model obtained the best results. As in the preceding project, the authors in [13] performed a comparison between SVM, Naive Bayes, Logistic Regression and Multinomial Naive Bayes models. In this study, the authors verified that they obtained the best results for their Corpora, with about 87% accuracy using the SVM model.

In [14, 15] the authors developed two models for the Portuguese Navy, both addressing the issue of classification of emails sent to this institution. The models were developed with the intent of reducing the manual labour time occupied by this activity, and to reduce the human error in this activity. Although both authors shared the theme, the author [15] applied more traditional models; using the Linear Regression model an accuracy of 82,5% was achieved. Author [14] applied Bidirectional Encoder Representations from Transformers (BERT) and Linear Support Vector Classification (SVC), obtaining an accuracy of 92%.

An outlier in the literature when it comes to implementing solutions for the situations raised above, regarding request classification, is the implementation of chatbots, which according to [16] consist of a software application based on Artificial Intelligence (AI) that simulates a conversation with a human. According to [17], this type of system automates help desk processes using a database of known problems and resolutions. Using the data previously described, a chat system is created in which the customer provides information, and the system tries to resolve it, after categorizing the given information and later developing a written text or full repost of a given response, based on the system library of resolutions.

The authors [18, 19], carried out a process like that described for the chat bot, mentioned in the paragraph above, but in their case the chat is initiated by the reception

of emails from their client rather than a short message conversation system, and their system was trained using the emails that were already available at the time of the construction of their model. It is important to point out that the author [18], who intends to generate automatic responses for his project, attaches great importance to the ability to build automatic responses. His conclusions emphasise the quality of the responses that this mechanism allows, where the rapid response to generic and less technical constraints is regarded as the greatest added value since it frees up help desk operators for more complex constraints. On the other hand, the author [19] describes his artefact in greater technical detail, where he verifies that classical and more "economical" methods such as a Multilayer perceptron (MLP) have very competitive results compared to the use of Deep Learning Networks, such as LSTM, more sceptically LSTM-G, indicating that if the training corpus has a more significant size and there are no constraints associated with infrastructure, models such as SVM and LSTM-G will obtain better results.

2.4. Implementation model limitations for this use case

When implementing these models, one constraint detected for these systems is that the quality of the information provided by the client can cause problems. The information provided by the user can vary significantly, in content and quality, as explored by [20]. That said, the same author also points out that when the information is sent by email, the text is often similar and tends to be small. Moreover, when help desk operators respond to these requests, they don't tend to respond with the full resolution, either for technical or business reasons.

In [21, 22] the authors detected a significant constraint in the distribution of data that had a large number of records associated with one or two categories. They used undersampling methods to attempt to overcome this constraint. This method consisted in the random removal of records in classes with more representation in the training process. Another method used by these authors was the oversampling of the less populated classes to increase their weight in the training vectors. Another attempt was a classification by phases: in the first phase the less represented classes are labelled, and subsequently the more represented classes. In this way there is no direct competition of less represented classes with the more represented classes.

Chapter 3 – Help desk classification

This dissertation focusses on the development of an add-on application to the current manual help desk system operating in the port. This add-on will automatically classify the category, group and subcategory of the request through the use of the “Subject” and “Body Text” of the request. We are going to use five machine learning (ML) models: SVM; decision tree; random forest; a LSTM and a BERT to identify which model produces the best results regarding the data set. To have a larger data set and one that is less sensitive to seasonal issues, the data set used was extracted from the actual help desk system for all 2022.

3.1. Manual Classification

As mentioned in the introduction, the request is opened after a claimant sends an email to the service desk email server, or a member of the service desk opens one manually for further discussion. We will be mentioning these methods as Email Incident Reporting (EIR) and Internal Incident Report (IIR).

In the case the EIR, after the reception of the email the service desk application creates a new request that has yet to be classified. At this state the request only has the information that was available on the email, such as: subject, body, sender aka claimant, and attachments. A member of the service desk must then enter the request that has just been created in and fill all the mandatory classification fields, such as group, category and can also fill the subcategory.

Regarding the case IIR, the service desk operator opens the request, hence he must fill in all information that was automatically filled completed in the case of the EIR request, as well as manually classifying the request.

3.2. Vectorization methods

For the text processing task there are a series of associated tools, both in the vectorisation of the tokens and in the network training models, as well as in text pre-processing, where we can see the use of methods such as the removal of stop-words and standardizing to lowercase. This consists of removing words that may have little relevance in the corpora, such as the word "in. All information is passed to lowercase, since for the computer the words "one" and "One" are two different tokens. The

aforementioned are two of the most common methods, but other can also be applied depending on the specificity of the project.

For NLP, the tokens considered are n-grams, and by definition the n-gram can consist of a unigram such as the word "test", or a more complex n-gram such as bigrams, for example "United States". The great advantage of applying n-grams greater than the unigram is the retention of the original meaning of the combination of some words in sequence in a single token, but it increases the complexity in the formation of the tokens [21]. After the process of tokenization, these must be vectorized in order to be comprehensible to the method, some of these methods are the ones described below.

3.2.1 Bag of Words

The Bag of Words (BOW) is one of the most simplistic ways of approaching vectorisation. This form of vector representation of tokens is based on counting the number of tokens that exist in a *corpus*. As verified by [23], although this form of vectorisation has interesting results with text pre-processing methods such as the removal of Stop-Words, the type of text pre-processing used for BOW must be selected considering the challenge at hand. It should be noted that in the review by these authors, it is stated that in similar challenges and using models such as Maximum Entropy, SVM and Naïve Bayes, accuracy rates of 90% have been achieved.

3.2.1 Term Frequency – Inverse Document Frequency (TF-IDF)

This is a way of assigning weight to a token present in the *corpus* in the vectorization process. For this calculation, the number of times the token occurs in a given record is taken into account, this being the Term Frequency section, and then the *corpus* is checked to see how many times the token occurs in other records, this being the Inverse Document Frequency section. [21] The token will be assigned a weight by checking the number of occurrences it has in a record. However, it will lose out if it is found in many documents.

3.3. ML models

Due to the nature of recorded requests, which are usually recorded in written format [21], many of the solutions found in the literature use more, or less, sophisticated Machine Learning methods, many of which still opt for more classical models, neural networks, among others. However, there is a growing use of tools associated with Transformers, which are the latest major revolution in this area of knowledge since the release of the article “Attention is all you need” [22].

Broadly these models can be divided into three categories: classical, neural networks and transformers. In the section below, we'll go into greater detail on the models most frequently verified in the literature in problems like the one related to this study.

3.3.1 Support Vector Machines

The SVM aims to separate tokens by maximising the distance between the hyperplanes of each class in an n-dimensional space.

In its execution, the SVM model finds a hyperplane of the class, the line separating two classes [24]. According to [25], this separation line is the line that maximises the minimum distance between the hyperplane and the examples of the closest classes. The greater the margin, calculated by doubling the distance between the support vector of a class and the hyperplane, the more comprehensive the model will be and the better it will be at classifying data that does not exist in the training data.

3.3.2 Naive Bayes

The Naive Bayes model is a probabilistic classification model based on Bayes' theorem. In order to calculate in which class the new record is categorised, the tokens of the new record are compared with all the tokens of the classes already trained, and it is placed in the class where the calculation obtained the highest value. [21]

3.3.3 Logistic Regression

Logistic Regression will associate a new record with a class based on the calculation of the probability of this record belonging to that class. For this calculation, the new record will have its weight calculated using the weights assigned to the representative tokens of the classes already trained.

To calculate this probability, the sigmoid function is used to calculate the weights of the tokens present, which are then multiplied by the weights of the new class tokens and the result is assigned a value between 0 and 1. The higher this value, the more similar the record is to the class being compared. [25]

3.3.4 Decision Tree

The Decision Tree algorithm is a machine learning technique that is commonly used in Natural Language Processing tasks. It works by recursively dividing the training data into subsets based on the features values, ultimately creating a tree-like structure of

decision nodes and leaf nodes that are the end of a branch. When a new input is provided, it traverses the tree to make a prediction or classification based on the learned decision rules. By analysing the features of the input text, the Decision Tree can make informed and accurate predictions about the content, context, and sentiment of the text.

The effectiveness of the decision tree algorithm in Natural Language Processing (NLP) lies in its ability to handle non-linear relationships between features and the target variable, as well as its interpretability, which allows users to understand and interpret the decision-making process. [26]

3.3.5 Random Forest

Also known as Random Decision Tree forest, Random Forest is an ensemble model made of many decision trees.

In a Random Forest algorithm, each tree is trained on a unique subset of both the data and features, enhancing diversity within the forest. This diversity helps in capturing different aspects of the underlying patterns in the data. Rather than relying on a single decision tree, the algorithm aggregates predictions from multiple trees by averaging their votes, leading to more robust and accurate predictions. This ensemble approach not only reduces overfitting but also improves the model's generalization capability [27].

3.3.6 Long short-term memory

The LSTM is a Neural Network (NN) based on the Recurrent Neural Network (RNN) architecture, which focuses on the capacity of the neuronal network to retain long-distance dependencies. This model was proposed by [28], and its great advantage for NLP problems that the LSTM brings, compared to the RNN, is the minimisation of the associated "Vanishing Gradient" constraint.

This architecture, when compared to a classic neuronal network, which has an activation value at the input and an activation value at the output, is known in the literature as ai and bi respectively, in this respect the LSTM has an architecture with three "Gates", these being the input, output and forgetting. In the input and output gates, the calculation for the node is carried out by the gates by adding the activation values of the previous hidden layer with the activation values of the current layer as well as the activation values of the LSTM.

3.3.7 BERT

BERT is a model with a special focus on the attention of the network, with a bidirectional transformer network, which is already pre-trained with an expansive *corpus*. This model is used to develop NLP solutions. Since this Transformer is already pre-trained, the effort made to adapt it to the specific problem tends to be made in its fine-tuning.

This Transformer works by representing the text in a sequence of tokens, and assigning weights to these tokens considers the sum generated in the token's embedding, as well as its position in the text.

There are various options for using BERT transformer, the most common being the basic version with 768 hidden layers and the model version (large) with 1024 hidden layers in the network [29]. It should be noted that if these versions don't achieve the expected results, there are a myriad of platforms that offer access to pipelines based on this transformer that may already be better prepared for a specific problem, an example of which is the Hugging Face portal.

3.4. GMP prototype

The prototype "Gestão Melhorada de Pedidos" (GMP) will pick up on the manual classification system and experimenting with the methods presented about, it will look to transform the manual system into an automated one. It will focus on the email incident reporting case, since in the internal incident reporting case it is expected that the user classifies the request correctly, since the request classification is creating by an operator who understands the language in the subject and body. Moreover, if there are problems the operator is guided by a senior help desk member.

In order to obtain the request information GMP will rely on the current helpdesk system to send the request via it’s REST API and once this data arrives one of the scenarios will occur. If the fields regarding group, category and subcategory are filled the GMP will assume that this data is pertaining to a training purpose and store it on its training data database. Otherwise, it will assume that receive request is to be classified and it will send it into the classification database.

The training purpose data will be used to train the selected ML model that. To ensure that the model retains it’s viability trough, a procedure will occur every week where GMP

will call service desk REST API in order to obtain all the requests that were closed during that week. Once the original is stored the data, will be apply the methods discussed on section 4, adding new columns the tokenized subject, body and claimant. In this format, GMP provides the operator with a centralized location where he can perform data analysis as well perform tweaks to GMP NLP models. Lastly GMP will verify if the request received on the weekly update have their request id present on classifying purpose table. If they are, it will compare the final labels with those in the classifying purpose model. This approach enables passive monitoring of GMP's accuracy metrics during operation.

Regarding classifying purpose model, after a new request is created in GMP, in the email incident reporting case, service desk will use the REST API to send this request to GMP, this will create a copy in GMP. Once this copy is created the data will be pre-processed using the methods discussed on section 4, and new columns will be added for the tokenized subject, body, and claimant. Once this process is completed GMP ML model will be process the data and attribute a label to classify the group, category and subcategory and once this step is completed GMP will send these labels to the service desk for the request to be classified on service desk and GMP will update the request in GMP with the same labels.

Since GMP is only a pilot at the time of this dissertation, the REST API from the helpdesk is not yet in use. Additionally, there isn't an allocated server for this project. Therefore, all experiments were conducted using a personal computer. All data requests were obtained using an export function available on the service desk application, which collects the filtered data into a CSV file. This process is reflected bellow in figure 2.

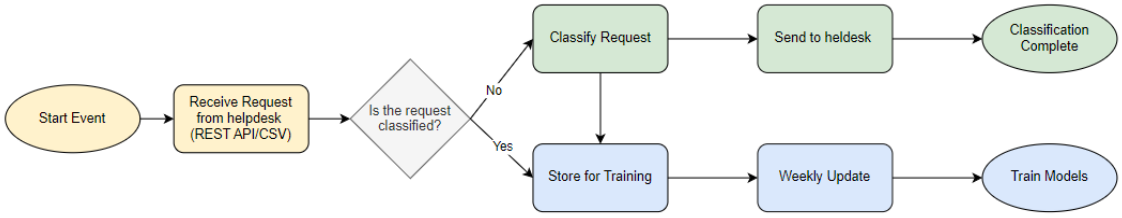

Figure 2 – GMP workflow

Chapter 4 – Dataset

The data set is reviewed employing a two-way approach. First, we will review how the data was extracted from the Help Desk, then we briefly analyse the data format and its distribution in groups and categories. Finally, we will examine the data cleaning process that was used during the current experiments.

4.1. Data Extraction

The selected data pertained to all the requests that were registered during the year of 2022 coming to total of 19897 formal requests. These requests are mainly written in Portuguese having only a few exceptions that are written in English or Spanish.

The report functionality native to the help-desk software was used to extract the data. The file with the extracted data is a CSV file containing the requests. It has the following columns: "Request ID", "Subject", "Requester", "Body", "Category", "Subcategory", "Group", "Creation Date".

As mentioned in the previous section, we mainly focused on using the Subject to determine the group and category of the request, and the request body text for group, category and subcategory classification. The rest of the columns were extracted for other future developments so they will not be further used or mentioned in the current dissertation.

The Subject, this column can generally have all sorts of characters but does not exceed the length of 32 words on this data set. In this column the Requester will briefly explain the motive of their enquiry, and depending on their background will resort to the use of coded sequences, such as a transport container plate and the respective applicational form on which analysis is required.

The characteristics described above are reflected on the entries of this column, which average 6 words with a standard deviation of 2.5 words, and a median of 5 words. As showed on figure 3, the quartile distribution indicates a first quartile of 3 words and the third quartile of 7 words.

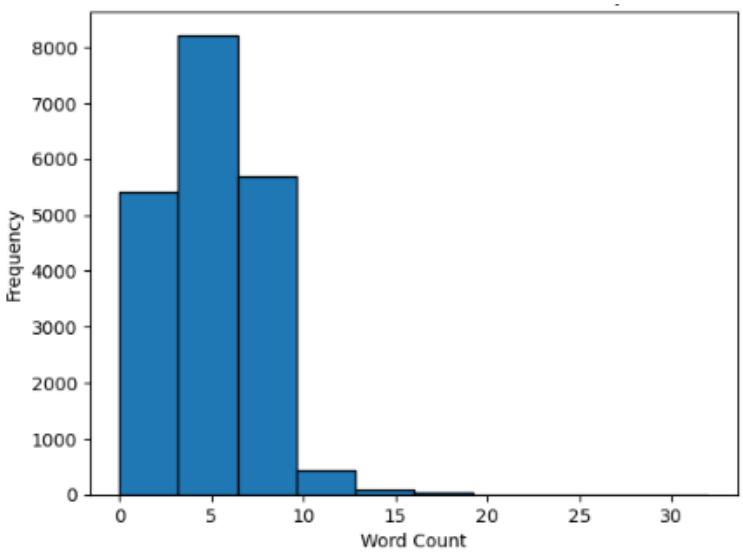

Figure 3 – Distribution of Word Counts in Email Subject

The request's body is where the requester details his incident in further detail and where the requester will try to add evidence and metadata, such as a container plate, an application name, in order to further describe the incident. This data can vary quite a lot between each request, since some of the requesters have templates that they use to create the request more easily, but some requesters put a lot on information that isn't relevant or erroneous. But on average the body has 205 words on each request and a median of 147 words and there is a standard deviation of 250. This deviation can be explained since that the third quartile ends with the body and the body contains up to 249 words. The first quartile ends with the 54 words. The biggest email has 2680 words, to be noted that almost all emails have a signature that will add up to 10 words to each email, or more if the requester's entity has displayed a message regarding privacy and confidentiality warning. On the other hand, in the body there can be images and tables that will be problematic while handling the data. This is distribution is also showed below on figure 4 as to give a visual aid.

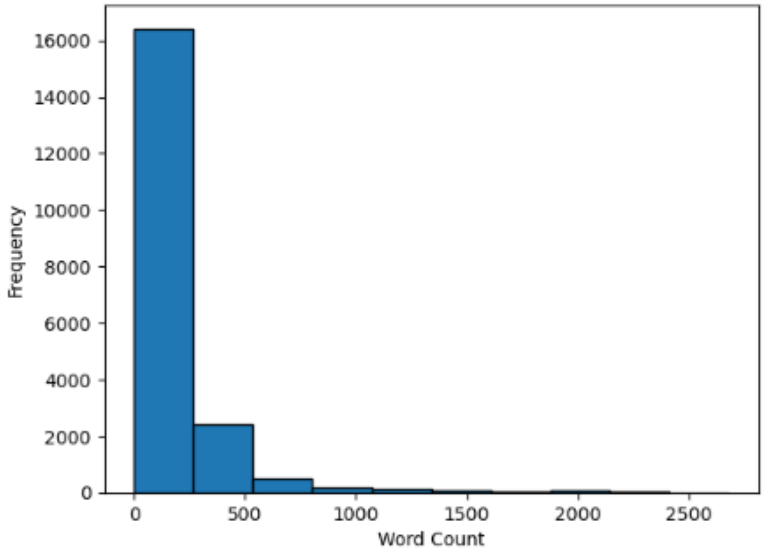

Figure 4 – Distribution of word counts in Email Body Text.

The "Group" column has the five work groups that encompasses the reality of the service desk operators. This work groups tend to be specific to the operators' training and their current workstation at service desk operation. It is rare that an operator belongs to more than one group. The groups in this paper will be identified as G1, G2, G3, G4 and G5.

The "Category" column consists of the 16 main fields of Operation where the service desk operators have direct intervention. In this dissertation these categories will be identified as A, B, C, D, E, F, G, H, I, J, K, L, M, N, O and P. These categories are ordered in descending order, and category A, B and C have 11713, 4655 and 980 requests respectively. The categories O and P have 18 and 14 requests respectively.

The "Subcategory" column has 68 labels, and these labels are fully dependent on the selected Category. Since there are 68 labels, we will mention them from SB1 to SB68 and they are arranged in descending order by the number of requests each label has. In this column the label imbalance is even more noticeable since when we have subcategory labels SB1, SB2, SB3, SB4 with 4648, 3407, 2074 and 2070 requests to quickly drop to 1078 and 647 requests for the subcategory labels SB5 and SB6. By the subcategory label SB28, that has 104 requests, all labels bellow have less than 100 entries and the last tree subcategory labels have 2 or 1 entry.

4.2. Data Preparation for Subject

We carried out a series of text pre-processing methods which we will now list in order of execution: removal of the "RE:" prefix commonly associated to the response of a previous email; removal of the "FW:" prefix commonly associated to the forwarding of a previous email; lower casing of all characters; removal of Portuguese stop-words, no development was made into English or Spanish stop-words since they are an exception on the current corpora; removal of special characters; various standardization of maritime var-char sequences regarding the maritime business that we will clarify in the paragraph below, and lastly removal of numeric digits.

The maritime var-char sequences are coded sequences that offer specific information in a strict and direct manner, such as the vessel call number commonly used by the Port authority, e.g., "PTSIE123022272". that can be translated as: "PTSIE" -» Locode for port of Call; "1" means of transport which in the example means maritime, "23" is the year, and it ends with a six-digit sequential number. In Table 1 we give a full set of examples of these codes, as well of their meaning and we prepared them for tokenization. To be noted that we have applied this process to reduce the number of what would have been unique tokens, in a series of tokens that are more transversal in all the Corpora.

Table 1 – Example of maritime var-char sequences.

Meaning	**Regular Expression**	**Conversion**
Sines Maritime Transport	ptsie1[\d]{8}	ptsiea
Bobadela Rail Transport	ptbbl2[\d]{8}	ptbblb
Portimão Road Transport	ptprm3[\d]{8}	ptprmc
Bill of Lading	[a-z]{5}[\d]{7}	bill_lading
Container Plate	[a-z]{4}[\d]{7}	matricula_cn
Customs Document	[\d]{2}[a-z]{2}[\d]{14}	doc_digitalizado

Finally, as we will further explore below, we noticed that some of the categories that were used by the operators did not correspond with the Group indicated. For example, this occurred with a person identification software that is employed by the Port Administration, that can have three categories, depending on which working group had to act upon the request (e.g. G1, G2, G3). This operation also involved some alterations

in the classification regarding the group, that proved to help the data unbalance a little. Also, there were two categories that were renamed, by the Port Administration helpdesk team directly on the Helpdesk system, during the year 2022 and as such there was the need to convert the previous nomenclature to the latest one in order to standardize the data.

In terms of Category request distribution of data is clearly illustrated in the last two columns of Table 2 where we can see that most requests belong to the categories "A" and "C" which have 11711 and 4655 requests respectively out of the 19897 total requests. A similar situation has occurred when it came to the Group request Distribution. As can be seen in the first two columns of table 2, the groups G1 and G3 received most of the requests.

In order to vectorize the subject, a standard TF-IDF Vectorization for all the models was used with exception the algorithms based on neural networks were specific vectorizers were used. A train and test split was applied using 20% of the data set as the test vector.

Table 2 – Group and Category request distribution

Group	Group count
G1	11717
G3	6424
G2	951
G5	281
G4	81

Category	Category Count
A	11711
C	4655
B	980
D	825
I	386
E	370
G	285
F	238
H	191
K	95
L	68
P	30
O	29
M	20
N	14

4.3. Data Preparation for Request Body

Since the second pilot was based on the body of the request, and we added a further label for classification, the subcategory, most of the methods described in the section above were also applied here.

Contrary to the subject, the request did not have the prefix “FW:” or “RE:” but will have a signature specific to each person and organization. Therefore, instead of removing these prefixes, a function was developed to remove everything from the sender’s signature at the bottom of the text. This step included removing the corporate image and dealing with the information security notice present in most of the corporative emails.

To test our assumptions regarding the text pre-processing the execution describe above will be used only on the Complete and Tokenized vectors. We will further delve deeper into to these vectors bellow, after table 3.

Regarding the distribution of the requests by Group and Category the values presented in table 2 are still accurate for this case, since the second pilot also classifies these fields. In addition, the field Subcategory was added. As mentioned above, there were some requests that were miscategorized and this added a direct correlation to the misclassification of the subcategory since when a category is selected by the operator, this will limit the subcategories present on the Help Desk system for selection. In addition, there were also several misclassified subcategories on the correct category. It is also worth mentioning that some categories don't have subcategories.

Since some subcategories don't have enough requests to be realistically used to train and test the algorithms, they were merged into SB6 which was the label created to describe no subcategory assigned.

Table 3 – Subcategory distribution by Category

Category	Category Count
A	SB2(3407), SB3(2074), SB4(2070), SB5(1078), SB7(628), SB8(505), SB10(425), SB11(307), SB12(291), SB14(218), SB17(186), SB18(168), SB21(139), SB32(64), SB36(42), SB38(32), SB46(17), SB47(16), SB49(12), SB54(8), SB55(7), SB56(7), SB59(7), SB64(4), SB68(1)
B	SB6(33), SB13(277), SB16(212), SB20(146), SB23(123), SB27(82), SB28(82), SB42(25)
C	SB1(4648), SB58(7)
D	SB6(257), SB9(452), SB34(54), SB44(24), SB45(18), SB57(7), SB61(6), SB62(5), SB67(2)
E	SB6(86), SB15(199), SB22(85)
F	SB6(27), SB24(116), SB39(31), SB40(30), SB41(25), SB52(9)
G	SB6(22), SB25(111), SB29(80), SB31(72)
H	SB6(37), SB30(73), SB35(44), SB43(24), SB48(13)
I	SB6(27), SB19(156), SB26(104), SB33(60), SB37(32), SB60(7)
L	SB6(68)
M	SB6(7), SB51(11)
N	SB6(8), SB63(4), SB66(2)
O	SB6(6), SB50(11), SB53(9), SB65(3)
P	SB6(30)
K	SB6(39), SB15(16), SB22(40)

When compared with the subject the body size for each request is much larger meaning that there was a hypothesis that more pre-processing methods could be advantageous. As such for each algorithm configuration we have used two more methods, lemmatization, and Part of Speech Tagging (POS Tagging) and have separated the dataset into 5 training vectors:

- **Complete** — It has all the preprocessing methods described in section 4, but beforehand it was applied POS Tagging and lemmatization, by this order.
- **Lemmatization** — The data was lemmatized then it was tokenized using the base TF-IDF vectorization with default configuration.
- **POST** — Before applying the base TF-IDF vectorization with default configuration the vector was POS Tagged and then vectorized.
- **Tokenized** — The data was only pre-processed according to what is described on the first two sections of this sub-chapter.
- **Without tokenization** — The data was directly inserted into the vectorization function without any pre-processing.

Regarding vectorization there was one subcategory that had to be abandoned since it only had one registry and as such it was changed into the generic label, SB6.

Chapter 5 – GMP Development

This section analyses the approaches that were used after the collection of the data and its preparation for model training and testing.

In this chapter we will be dividing the experiments into two main subsections that are the experiments using the request subject and the second section where the request body is used. This approach was used to execute the first experiments with a smaller sample of words for each request and to verify how effective it was. The second experiment was carried out on the full body.

The first architecture pertains to the classification of the group and category as independent targets, with isolated NLP models. While the second architecture classifies the group and category in a hierarchical manner that we will further describe below.

5.1. Request subject Category and Group classification

In this section since the subject by norm is small and the information condensed the text pre-processing methods were reduced to a minimum where we mainly applied the methods stated above and did not delve too much into fine tune each network.

5.1.1. Non-hierarchical models

For the first experiments the request subject was used to the train the models to predict the classification of the Groups and the Categories. The models used for this classification were SVM, SVM with artificial oversampling, Decision Tree, Long Short-Term Memory (LSTM) model and a BERT model.

We used the default function, from “Sklearn”, with exception of the gamma parameter set to auto in the SVM model. We also added the function One versus Rest, as this function will help this model deal with the unbalanced data set, since it fits each class to the model while comparing it against all other classes, doing this one class at the time. We used this SVM configuration to test artificial oversampling, using the smote function from the “Imblearn” library. We have applied this function to the train and test vectors for this model, since this will inflate the values present on the less populated classes, to the same number as the biggest class.

In the Decision Tree model, we used the entropy criterion, the max depth of three and the random state of 0. For the Random Forest Model, the model was set to have 400

estimators, entropy criterion and a random state of 0. These parameters where chosen as they were commonly used on other models that we saw on the literature.

Lastly, experiments with the Naive Bayes algorithm and Logistic Regression from the “Sklearn” library were carried out to verify how they would handle this unbalanced dataset. Still, these experiments were made without delving into the hyperparameters where only the default values for the mandatory hyperparameters were used.

We collected the LSTM model, from the “Keras” library. I had an embedding layer with 250 neurons, followed by a spatial dropout layer of 0.2. After these two layers the network arrives at the LSTM layer which is followed by a final layer with the same number of neurons as there are classes, e.g. 15 for Category and 5 for Group. The number of epochs was set to 100, but we also applied an early stop function with 10 epochs of tolerance. In general, for the category LSTM model the recurrent number of epochs was around 16 and for the group classification around six epochs.

The final model we used was the BERT base uncased model from the “Torch” library. Since it is a BERT model, we also used a tokenizer from this library and the respective label encoder and tensor. Regarding the model configuration we used the batch size of 16, the Adam optimizer and the learning rate of “2e-5” and 8 epochs for training.

5.1.2. Hierarchical models

In terms of Category classification, when executing the models of the previous section we noticed that category "A" had a high number of false positives. There was also a dispersion of classifications regarding categories "B", "K" and "A", since these three Categories share the same core business, but pertain to different areas of resolution. The Group classification was also affected but to a lesser extension. The models in the previous section operated in an "all classes versus all classes" fashion for category classification. We consider functionality could be improved if the subject of the request was classified by Group first and depending on this result: Category to reduce the number of classes in each category model.

Using the business rule that all categories aren't available for all service desk operators, and the selection of categories are highly dependent on the group of operators, we have developed a new model that uses a hierarchical architecture to classify the category, following the structure depicted in Table 4.

Table 4 – Group and Category Corelation

Group	Category correlation
G1	A, M
G2	B
G3	O, P, N, M, L, C, D, E, F, H, K
G4	F, K, D, E, G, H, N
G5	I, G, P, M, E, K

In this model, the first step was to separate the data set into two subsets, one for training with 15918 requests (80%) and another with 3979 requests (20%) for test. Subsequently the SVM for Group classification model, as described in section 5.1.1, was trained using the train data set and the predictions from this model were added to a new column on the test data set denominated "Predictions". From this point forward, train and test data sets were again split depending on which group, they were labelled as in the column "Group" for the train data set, and the column "Predicted_Group" for the test data set. Each train-data set was used to train a group specific SVM model. The SVM model that we used had the same specs as the one described in the previous section, except for the train and test split function. After the generation of each model, the Group specific data set for each group was tested, and the predicted values were added in a new column "Predictions_Category". The only exception was the Group "B", since this Group has only one category and as such the value of the category "B" was simply applied to the column " Predictions_Category" of its test data set. Finally, all the test data sets were merged, thus giving valid metrics for this model. This architecture was also applied using the same Decision Tree, Random Forest and BERT models with the same specs as described on section 5.1, but always having the group classification executed by the SVM model.

5.2. Request's Body Subcategory, Category and Group classification

5.2.1. Non-hierarchical models

When compared with the previous experiments we have dropped the SVM algorithm with artificial oversampling since the results during the first experiment didn't develop an improvement in classification, since it would improve the classification of the less

populated categories and groups but would significantly compromise the accuracy when regarding the bigger categories and groups, at least when nominal values are considered.

Another significant alteration was the implementation of the function “GridSearchCV” from Sklearn library to auto determine the best hyperparameters for the algorithm classical algorithms (SVM, Decision Tree and Random Forest). This means that for each classification experiment (group, category, and subcategory) the hyperparameters could be different, and this improved the results for some of the algorithms. On Appendix A, C and E these configurations are declared.

Naive Bayes and Logistical Regression were dropped since they didn’t prove to be better at classifying the data, when compared to the SVM and Random Forest, additionally we used the Decision Tree algorithm as baseline to see if the grid search function can significantly improve the results.

A similar approach was attempted for the definition of params for the vectorizing function, yet the computational resources available didn’t allow the achievement of results with this method in a predictable and viable fashion.

Regarding the LSTM experiment we also used the same training vectors approach, but when compared with the configuration described in section 5.1.1, we reduced the number of patience epochs in the early stopping function since each epoch would considerably need more time to execute, and as such we reduced it by half to 5 epochs. We also did an experiment with auto-determination of layers, yet by similar constraints as the ones referenced for the vectorization function for the classical models these experiments didn’t develop results.

Lastly the experiments with BERT remained mostly unaltered since BERT has a specific vectorizer that already applies some of the methods we already were implementing on the other algorithms. The only difference was it classified the subcategory as SB6, transversal subcategory used for requests that don’t have an inherited subcategory, when there were less than 4 total entries for that subcategory.

5.2.2. Hierarchical models

When applying the hierarchical approach, we reduced again the number of algorithms and only SVM and BERT were used, since when using the subject, they showed the most

promising results, and since the object was now the request body each registry would need more computational power and time.

Another experiment was applied using a mixed approach where we observed which algorithm was best classifying the group of the request, and used the results to test the category, and then we applied the same logic to the subcategory models. This means, for example, that we could have the SVM algorithm classifying the group of the category, then the Category being classified by BERT and then have the subcategory being once again classified by SVM models.

Regarding SVM we have once again used the param grid function to determine which parameters were better suited for each algorithm, and we have also added the layer for subcategory classification that would have an algorithm for each category, meaning that on the third layer there were 16 algorithms configured.

One major concern that this approach had was the misclassification of a subcategory or category on the previous layer, meaning that if the group is misclassified the category will certainly be misclassified and if a category is misclassified the subcategory will most likely be misclassified. To be noted SB6 is a transversal subcategory used for requests that don't have an inherited subcategory, and as such we had to develop some escape functions in order to avoid this constraint. This function counted the number of available labels on the initial train and test vectors, still on the first classification layer, group classification, regarding category and sub-category to ensure that there weren't sub-categories and categories that were present on one vector and not the other.

There was also added a validation process during the execution of the third layer, subcategory classification, that was to verify if there were still labels available for testing. That is if a class had few registries, a case could happen when the label jumped to the second layer, it was no longer available for classification due to misclassification by the previous label classification. We found these cases, where the label was present on the training vector but not the test vector, and we would classify the result column of the test DF as SB6.

Chapter 6 – Simulation Results

In this section we present two different simulation results: results for Group and Category Classification as indepents models, and results for the hierarchical architecture. This approach allows a deeper analysis of the architecture of each model.

When deciding the metrics to be used recall, precision and F1-Score where considered. Precision can be described as the ratio between correctly predicted, true positives, tags and the total of predicted tags by the model, true positives plus false positives. On the other hand, Recall is the ratio between correctly predicted, true positives, tags and the actual number of tags that were available for that class, true positives plus False negatives. Lastly, the F1-score is a harmonic average between Recall and Precision, since in its formula it combines both concerns of Recall and Precision that it is particularly important when tackling an unbalanced dataset [30].

For our model analysis, we have mainly used F1-Score to compare results and to determine next steps, yet we also observed the Precision and Recall metrics in order to get a better understanding of the models' performance, yet they won't be used for detailed analysis and description of the models.

6.1. Request subject Category and Group classification

6.1.1. Non-hierarchical models

Regarding the group classification as can be seen in table 3, the model with best results was the SVM model with an F1-Score of 94.36% followed by the BERT model with 93.92%. The SVM model with artificial oversampling, Random Forest model, The Logistical Regression Model, the LSTM model and the Naive Bayes model had the F1-Score of 93.56%, 93.36%, 93.14%, 93.11% and 90.94% respectively. The Decision Tree Model was the weakest model with the F1-Score of 77.47%. In general, the Group models had better metric scores, when compared to the Category model.

Table 5 – Email Subject non-hierarchical model results for Group classification.

Model	Precision %	Recall %	F1 – Score %
SVM	94.49	94.57	94.36
SVM (art. oversample)	94.15	93.22	93.56
Decision Tree	76.51	81.71	77.47
Random Forest	93.70	93.32	93.36
Logistic Regression	93.53	93.57	93.14
Naive Bayes	91.27	91.68	90.94
LSTM	93.24	93.02	93.11
BERT	94.05	94.17	93.92

The following results are the results for the category models, as described in subsection 5.1, and are available in table 5. It was also visible that the SVM model has the best results with the F1-Score of 92.48%, followed by the BERT Model with 91.88%, the Random Forest model with 91.65%, the LSTM model with 90.84%, the SVM model with artificial oversampling with 88.75%, Logistical Regression model with 88.52%, Naïve Bayes model with 85.10% and lastly the Decision Tree model with 73.93%.

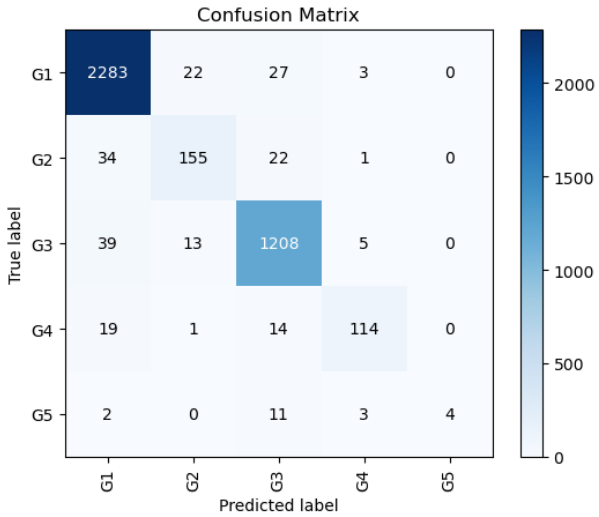

Figure 5 – Email subject SVM model for Group Classification non-hierarchical, confusion matrix.

Table 6 – Email Subject non-hierarchical model results for Category classification.

Model	**Precision %**	**Recall %**	**F1 – Score %**
SVM	92.35	92.84	92.48
SVM (art. oversample)	91.29	86.83	88.75
Decision Tree	70.99	81.66	73.93
Random Forest	92.44	92.44	91.65
Logistic Regression	89.38	90.03	88.52
Naive Bayes	85.75	87.74	85.10
LSTM	91.13	90.68	90.84
BERT	91.25	91.55	91.88

Regarding the experiments in this section, in general the major concern was the distribution of the predicted request. Groups "G1" and "G2" had a significant number of false positives since these groups were equivalent to roughly 90% of the data set, has seen on seen on figure 5. This outcome was even worse with the category classification where categories "A" and "C" corresponded to around 85% of the requests in a total of 15 categories. Regarding the category classification another situation of concern, as mentioned in section 5.2, were the Categories “B", "K" and "A" as they pertain to the same application but have different resolutions depending on the Group, this concern is visible on figure 6. As the subjects of these requests are quite similar the system had difficulties dealing with these requests.

Regarding the experiments with SVM with an artificial oversampling, it was noticed that even though the number of false positives for the major groups and categories decreased, the number of false negatives had a major increase. This is especially evident in the category prediction when comparing the SVM with the SVM with oversampling models recall metric. The values were 92.84% and 86.83% respectively, meaning that the major classes lost a lot of requests, and this hurts the overall prediction results.

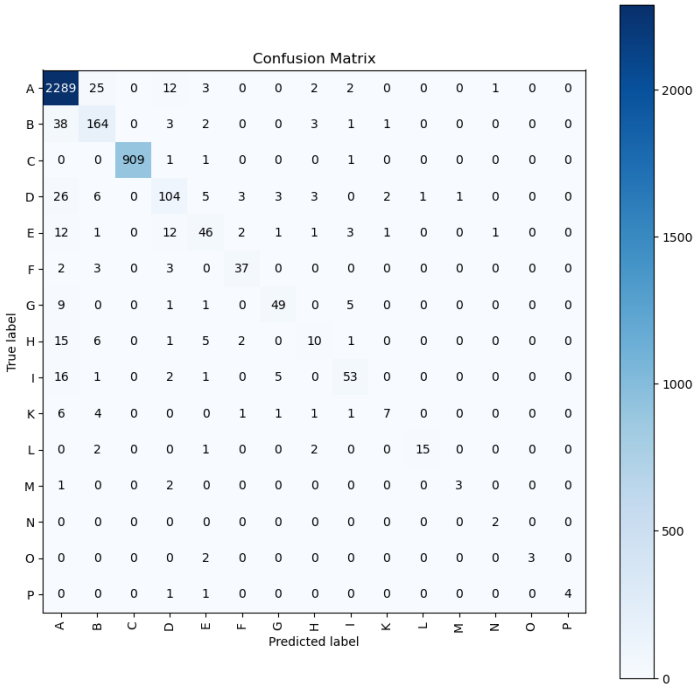

Figure 6 – Email subject SVM model for Category Classification non-hierarchical, confusion matrix.

A major advantage of the more classical models (SVM, Decision Tree and Random Forest) was the fast-training time. In general, this was accomplished in less than 10 minutes. On the other hand, the LSTM model needed around one day to train and the BERT model two days, being more resource intensive. To be noted that these models were trained using CPU with 32 GB of RAM.

6.1.2. Hierarchical models

While observing the results for these models in table 7, it is noticeable that no models were made using the LSTM and SVM with artificial Oversample models. We didn't use the LSTM model since previously it had not performed as well as the BERT model and they have the same degree of complexity when setting them up for this architecture. The SVM with artificial oversampling was dropped since it wasn't returning the expected results and increased the training time exponentially.

The F1-Score for these models were 90.60% for the Decision Tree model, 92.98% for the Random Forest model, 93.41% for the SVM model and the result of 92.04% for the BERT Model. Regarding the results obtain by these models, we can clearly observe that even though the Decision Tree model has once more the weakest results, they have been improved from 73.93% to 90.60%. The BERT model only had a marginal improvement

of less than 0.5%. Regarding the other models, the Random Forest model and the SVM model demonstrated an improvement of around 1% when compared to the category models discussed on section 5.1.

A major contributor to this situation was the fact that the categories "B", "K" and "A" were no longer in direct competition with one another, and that the categories "A" and "C" were now compartmentalized in their own group, diminishing the number of false positives that these categories had previously exhibited in the previous section. This is evident on figure 7.

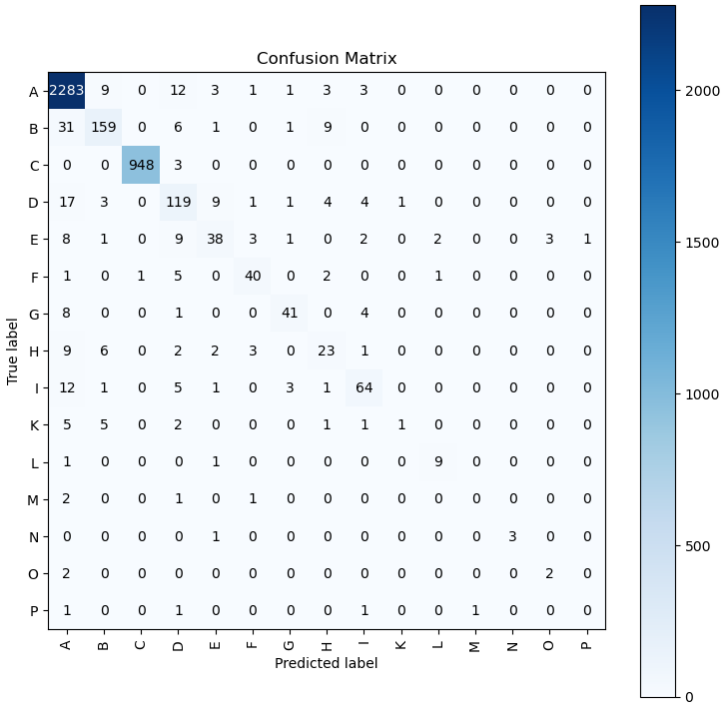

Figure 7 – Email subject SVM model for Category Classification as Hierarchical, confusion matrix.

One of the weaknesses of this architecture is that the category classification results are highly dependent on the results for the SVM model for group classification, since in this classification if a request that should have been classified as one group was classified as another, the following category classification is compromised.

Table 7 – Email Subject hierarchical model results for Category classification.

Model	Precision %	Recall %	F1 – Score %
SVM	93.29	93.74	93.41
Decision Tree	90.25	91.83	90.60
Random Forest	92.72	93.39	92.98

6.2. Request's Body Text for Subcategory, Category and Group classification

6.2.1. Non-hierarchical models

Since in the experiments executed using the request's body also included the use of different vectorization methods, such as lemmatization, post tagging, and others there was a major increase in results, that went from 12 experiments, six for group classification and six for category classification, with this new methods this has increased to 21 for each phase of classification, meaning that excluding the hierarchical model there were made 63 experiments, 21 per phase.

As such we will not discriminate each experiment and will only mention the best result per algorithm in each classification layer.

Regarding the Group classification the best algorithm was the SVM model which with the base vectorization achieved the F1-Score of 95.42%, and on figure 8 we can see it's confusion matrix. Yet to be noted that with the other vectorization method the lowest score was of 94.61% that is higher than the best result of the weaker algorithm.

The BERT classifier was a distinct second-best classifier once more, with the score of 94.87% followed by the algorithms of Random Forest, 92.78% with tokenized vectorization, Decision Three with 92.76% and having the base vectorization, and lastly LSTM with 91.77% with a tokenized vectorization. On table 8 we can see the precision and recall metrics regarding this experiments.

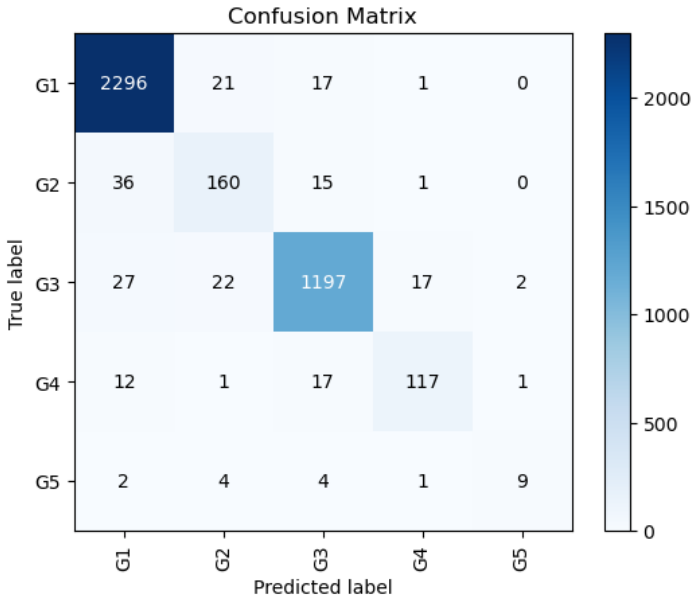

Figure 8 – Email body text classification SVM model for Group Classification non-hierarchical, with base vector, confusion matrix.

Regarding the vectorization, the best result on each algorithm was achieved while only using the base vectorization, or the tokenized approach, with a major exception of LSTM where the worst result was achieved with the base result. On the other side of the spectrum the tokenized and pos-tagging vectorization yielded the worst results, and the complete tokenization was the only one that didn't obtain the best or worst result.

Table 8 – Email Body text non-hierarchical model results for Group classification.

Model	**Precision %**	**Recall %**	**F1 – Score %**
SVM	95.43	95.50	95.42
Random Forest	93.06	93.19	92.78
Decision Tree	92.70	92.86	92.76
LSTM	91.75	91.98	91.77
BERT	N/A	N/A	94.87

For the Category classification the best result was also achieved by SVM using the base tokenization with the F1-Score of 93.47, with its confusion matrix on figure 9.

Regarding this algorithm the worst result was obtained with the complete tokenization with the result of 91.97%, as seen on table 9. Regarding the other algorithms the results in terms of stronger algorithms for this dataset maintained unaltered, having BERT achieved once again was the second-best result with 92.33%, followed by Random Forest with tokenized vectorization with 90.24%, followed by Decision Tree with 89.12 and lastly LSTM with 88.83%.

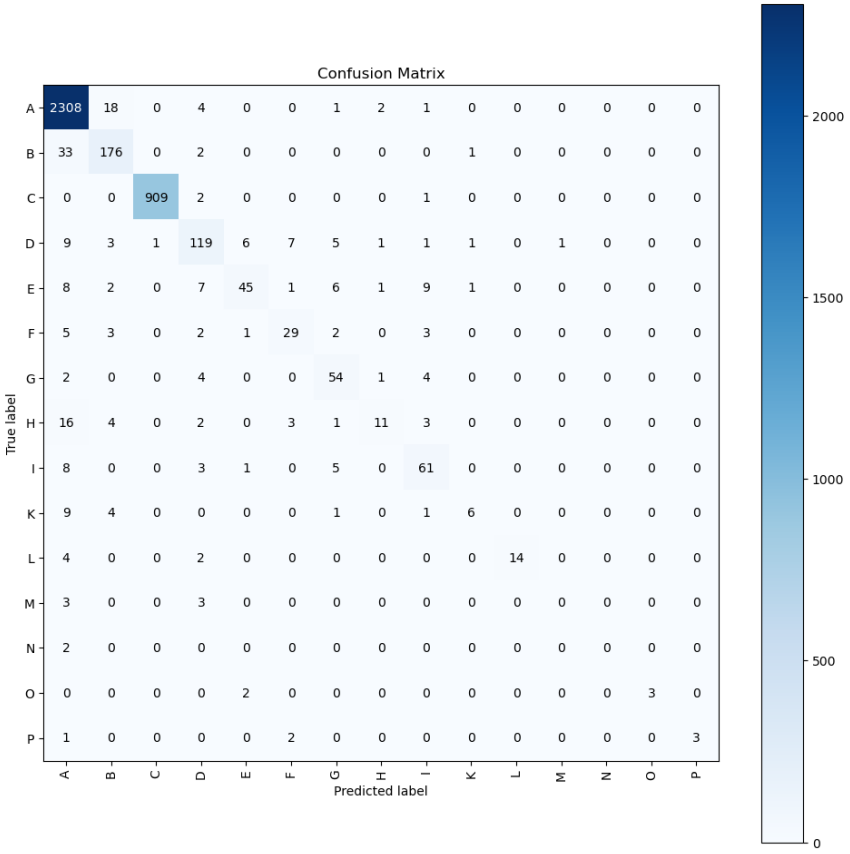

Figure 9 – Email body text SVM model for Category Classification non-hierarchical, confusion matrix.

To be noted that the terms of vectorization we now see that the base vectorization, with the exception LSTM and Random Forest obtained the best results using the base vectorization. With LSTM the best result was with the lemmatized vectorization and Random Forest had the best result using the Tokenized vectorization. Also, to be noted that this time the complete tokenization, with two worst results, had the worst results.

Table 9 – Email Body text non-hierarchical model results for Category classification.

Model	Precision %	Recall %	F1 – Score %
SVM	93.48	93.92	93.47
Random Forest	90.41	91.13	90.24
Decision Tree	89.19	89.30	89.12
LSTM	88.21	89.92	88.83
BERT	N/A	N/A	92.33

When observing the subcategory results on table 10, the best result was once again achieved by the SVM algorithm using the base vectorization having the F1-Score of 70.90%. Yet when it comes to the second-best classifier BERT and Random Forest share the second-best result, with the score of 70.31% using the base vectorization. Regarding the trends observed above Decision Tree algorithm, with base vectorization, was followed by LSTM with the tokenized vectorization, that had the classification of 65.71% and 60.78% respectively.

In this experiment it is noticeable that the results were all under 90% and the best result was only marginally above 70%. This is due the high number of classes to classify and how unbalanced the last classes were with some classes being too generic, such as class SB6, or being an overarching class such the class SB2 that is related to application communication and as such as similar or near identical classifications as the classes e.g. SB8 and SB11. This is visible on confusion matrix present on figure 10.

Table 10 – Email Body text non-hierarchical model results for Subcategory classification.

Model	Precision %	Recall %	F1 – Score %
SVM	71.48	71.63	70.90
Random Forest	67.38	67.14	65.71
Decision Tree	64.23	64.52	64.02
LSTM	60.98	62.26	60.78
BERT	N/A	N/A	70.31

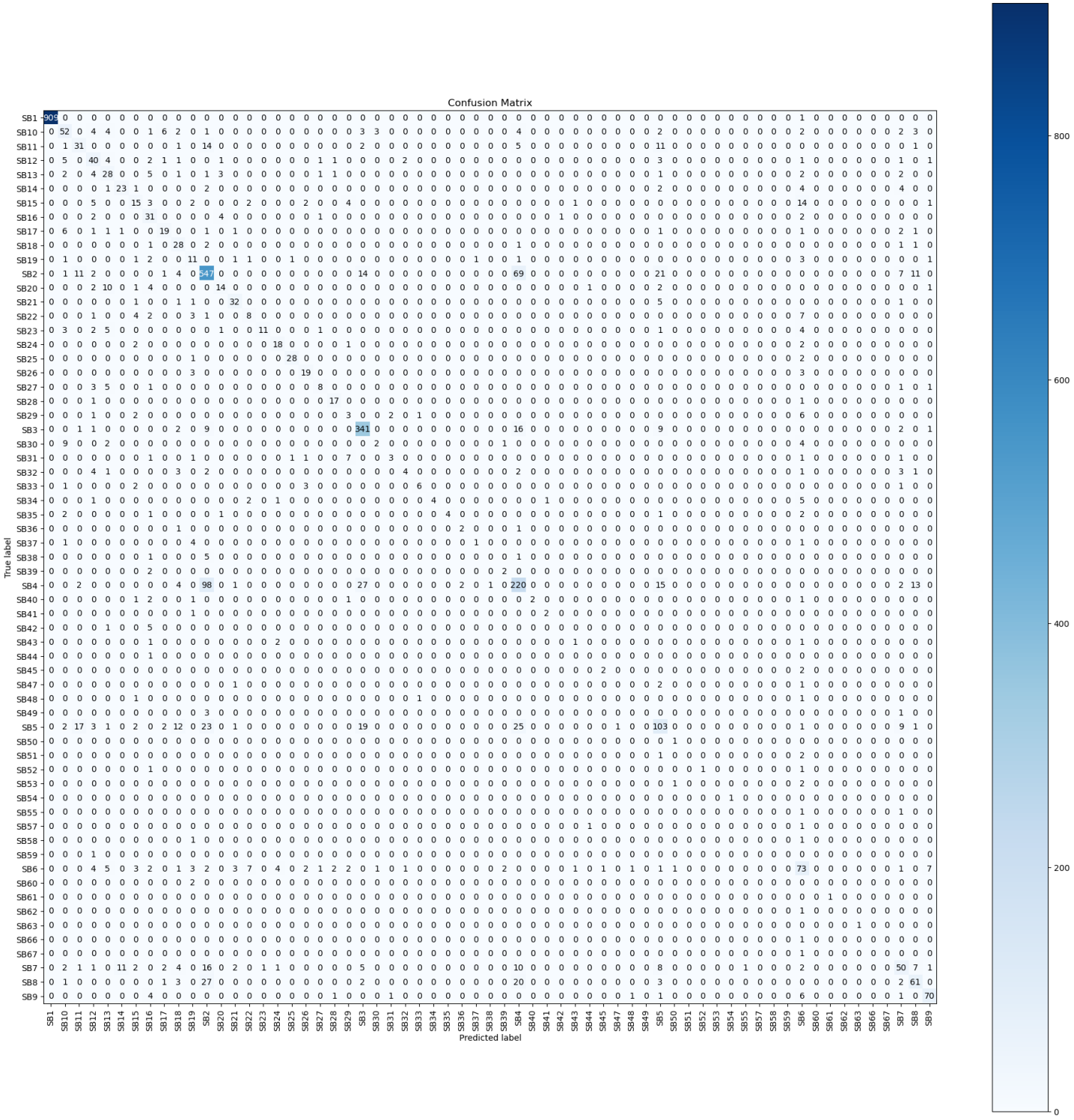

Figure 10 – Email body text SVM model for Subcategory Classification non-hierarchical, confusion matrix.

As an overview of the results described above, there was an improvement in the results in Group classification, having an improvement of 1.08% when compared to SVM model

using the subject. When observing the Category classification when comparing the SVM using the request's subject versus the request's body there is an improvement of 0.99% in F1 Score, yet this isn't as evident when comparing the result using the subject on the Hierarchical model, where the improvement of the classification was only 0.06% being marginal at best.

Table 11 – Email Body text non-hierarchical models summary results.

Model	**Group (F1 – Score%)**	**Category (F1 – Score%)**	**Subcategory (F1 – Score%)**
SVM	95.42	93.47	70.90
Random Forest	92.78	90.24	65.71
Decision Tree	92.76	89.12	64.02
LSTM	91.77	88.83	60.78
BERT	94.87	92.33	70.31

A major improvement in the results can be also due the implementation of the function "GridSearchCV" from Sklearn library in the classical algorithms that has helped the configuration of these algorithms to the models, and this is clearly visible when observing the results of the decision tree models.

While analysing some of the results in subcategories that had worst results, these tended to be the subcategories that had very few requests to train the model, and also some of these requests also suffered from a very small corpus, meaning that most of these requests where originally detailed via a phone communication, and posteriorly by opening request that served more as a formality that as a way of transmitting information to the helpdesk teams. In order to overview these results in a aggregate faction we have grouped the F1-Score results of this experiment on table 11.

Lastly to improve the results for subcategory classification, a major revamping would be needed in the number of subcategories. Subcategories where there where less than 10 requests should be evaluated, to verify its necessity. Alternatively, the algorithm could simply be trained to transform the values directly into class SB6. This would alleviate the number of classes that don't have enough expression for viable classification. Another measure would be revisiting the concept of class SB2 since it is encompassing business

areas of adjacent subcategories there is a large possibility of it being misused or of being redundant with other classes.

6.2.2. Vectorization experiments with non-hierarchical models

Regarding the experiments with the vectorization, and using the group results as the sample, we observe that SVM and Decision Tree had the best result when the base vectorization was applied- On the other hand LSTM and Random Forest preferred the Tokenized vector. These results may be viewed on table 12

In General terms the Pos-Tagged and complete vectors didn't assume any leading roles on the executed experiments. That may be attributed to the noise they may have caused to the model, especially the Complete vector since it had the most attributes for each token. This may have been due to increase of dimensionality that brought some sparsity in the tokens available that can lead into a quicker overfitting of the model due the increase of specialization regarding each class and will reduce the generalization. This may be evident when comparing the confusion matrix of SVM with the base vector, present on figure 8, with the confusion matrix of SVM model with the complete vector that is represented on figure 11.

The reduction in noise brought by the lemmatized vector and tokenized, one that reduced and concentrated the token pool by semantics and the other by the business side, helped most algorithms to produce better results, with a better result being attributed to the Tokenized Vector that managed to be superior method with two algorithms. The reduction of individual tokens and increase of attributes didn't prove to be beneficial to the Decision Tree and SVM models, where the reduction of tokens may have led to a loss of some discriminative information and may have introduced some noise or distort some patterns regarding the token distribution on each class. On the other hand, on this model the increase of information on each token.

Table 12 – Email Body text vectorization model results for Group classification.

Model	Complete (F1-Score %)	Lemmatized (F1-Score %)	Pos-Tag (F1-Score %)	Tokenized (F1-Score %)	Base (F1-Score %)
SVM	94.27	94.61	94.36	94.62	95.42
Random Forest	90.64	91.92	90.11	92.78	92.38
Decision Tree	90.05	91.38	88.61	91.37	92.76
LSTM	90.68	91.72	90.80	91.77	90.65

When the experiments were executed for the classification of categories, a similar result was obtained, where SVM and Decision tree performed better with the base vectorization and LSTM and Random Forest with the Lemmatized and Tokenized vectors respectively. With this we see that the same algorithms preferred the same vectorization methods with a minor alteration where Category switched the best performance from the tokenized vector to the Lemmatized vector.

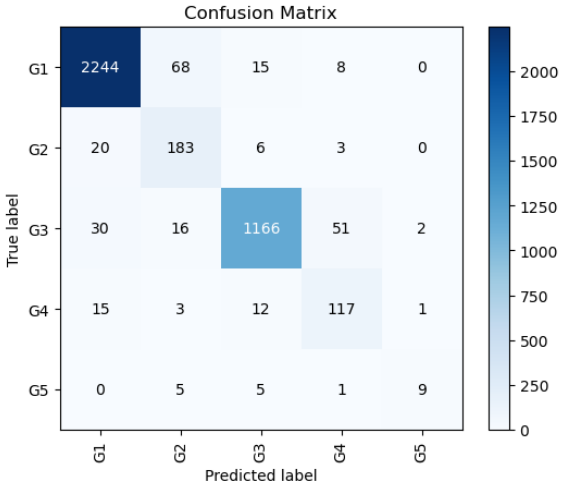

Figure 11 – Email body text SVM model for Group Classification non-hierarchical, with complete vector, confusion matrix.

Lastly in the experiments for subcategory classification an interesting change was that the complete vector had the second-best result when applied to the SVM model, that could imply that with the major increase in the number of labels and the general imbalance the vectors that had more attributes per token would improve, yet most of the worst results happened with the pos-tagged vector. Also, to be noted that all the models had better

results with base vectorization, with exception the LSTM model that had the worst result and instead had the best with the tokenized vector. Meaning that a simpler tokenization method, with exception of the LSTM model, resulted in a better training vector.

As an overview of dissertation experiments, with the nominal values present on table 13, the base vector has demonstrated that for this dataset was the most consistent results, and the vectors that have applied most of the features per token had in general the worst results. These experiments suggested that due to the current dataset, and dealing with the class imbalance, the applied models preferred a simpler feature representation, that may have helped the models better focus on the most discriminative features to help to deal with the unbalance. Experiments with other vectorization would be advisable to further investigate this matter, such the use of BOW instead of TF-IDF, further exploration of business specific general tokens. Yet the focus should be the treatment of the class imbalance on future experiments, since the results consistently get worse with the increase of labels to classify, despite the different experiments on vectorization.

Table 13 – Email Body text vectorization experiments with SVM model.

SVM Model	**Complete (F1-Score %)**	**Lemmatized (F1-Score %)**	**Pos-Tag (F1-Score %)**	**Tokenized (F1-Score %)**	**Base (F1-Score %)**
Group	94.27	94.61	94.36	94.62	95.42
Category	91.97	92.29	91.98	92.19	93.47
Subcategory	69.08	68.69	68.62	69.04	70.90

6.2.3. Hierarchical models

As mentioned in section 6.1.2 the results on these models will always be limited with the result of the layer before, meaning that if the F1-score on group classification is 95% then the category classification will only be equal or lower that 95% since the requests are already misclassified. There is an exception to this rule that occurs on subcategory classification since SB6 has a lack of classification and as such can be transversal to all categories. This situation had a very limited expression on the results.

Commencing with the model that used only the SVM algorithm, this algorithm started with the group classification F1-Score of 93.07% in category classification, illustrated in figure 12, and 71.90% for subcategory classification. With this result in category

classification, we can observe that contrary to section 6.1.2 the hierarchical model has developed a worse result than the best Category classification using as independents model for category classification. To be noted that when observing the results obtained the main difference was the distribution of requests on the test vector, that has slightly influenced the results. The model as emerged as slightly superior when classifying the subcategory with 1% better results.

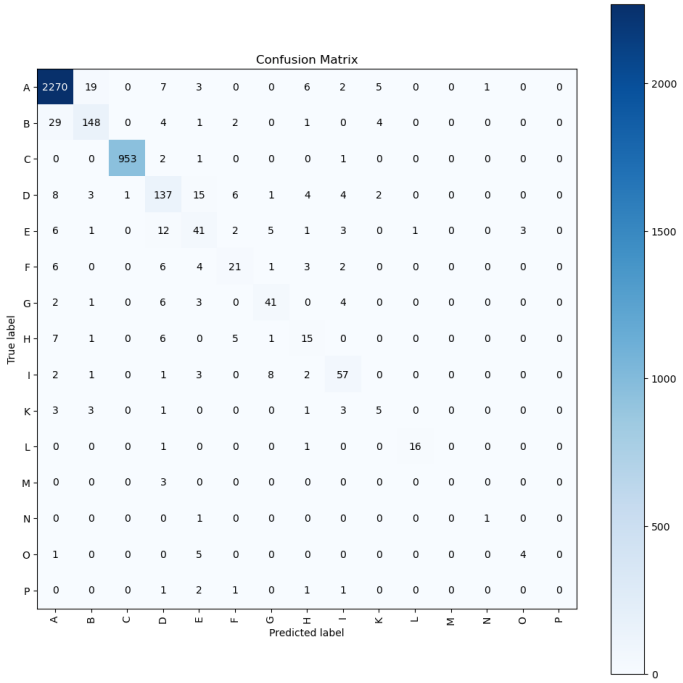

Figure 12 – Email body text SVM model for Category Classification as Hierarchical, confusion matrix.

Regarding the BERT model, it has the group classification F1-score of 94.87%, the category classification of 92.68% and the subcategory classification of 76.95%. When comparing the models, and knowing the handicap of starting with a worst result in group classification, when comparing the BERT hierarchical to the SVM hierarchical model is a large constraint for the result category classification yet it had a very small augment loss from one stage to another when compared to the SVM model but was superior slightly superior when compared to BERT model as independent, that achieved 92.33%. Yet when compared the subcategory classification we can see that BERT achieved better results than SVM by quite some margin, five percent. An interesting view of this data, when compared to BERT as independent subcategory classification was that this model had a better performance by filtering out irrelevant subcategories and focusing solely on those directly associated with the same overarching category, Hierarchical

BERT subcategory classification demonstrated enhanced accuracy and efficiency. This targeted approach not only optimized the model's predictive capabilities but also minimized the impact of noise, resulting in superior performance metrics.

Lastly the hybrid model used the SVM group classification, and as such started with the 95.42% F1-Score. using this a baseline both SVM models and BERT models were used for this dissertation results and BERT emerged slightly ahead with the score of 93.23%, that yet it is slightly worse than the SVM as a independent category classification. Lastly the subcategory classification achieved the best result with BERT models with the F1-Score of 77.93%. This last result even though it is quite underwhelming, it is a large improved when compared with all the other models having an improvement of almost 6% when compared to second best subcategory classifier. We can see a summary of these results on table 14.

When comparing the results of this section with the section 6.1.2. it was very noticeable that the new vectorization methods, the "GridSearchCV" function, for algorithm hyperparameters selection, and the corpus of each request has played a major role in these results and this model mainly maintains its advantages on subcategory classification where BERT was better able to deal with the unbalanced dataset when each model had a limited scope, when compared with the full dataset.

Table 14 – Email Body text model results for hierarchical classification.

Model	**Group (F1 – Score%)**	**Category (F1 – Score%)**	**Subcategory (F1 – Score%)**
SVM	95.42	93.07	71.90
SVM (group) | BERT	95.42	93.44	77.23
BERT	94.87	92.89	76.95

Yet the most interesting result was that the category classification using the subject on the SVM hierarchical model, achieved a slightly better result. Not having the same attention on the vectorization and the SVM hypermeters configuration.

6.3. General conclusions

The results produced with the best models for Group and Category classification with the F1-Score of 95.42% and 93.47%, have demonstrated that for this task these models are viable for further development. Yet when regarding the subcategory classification with the best F-score of 76.95% further development would be necessary to develop the model into a viable product.

The class unbalance has proven to be a major pain factor for the development of this models, and it was one of the main contributors to the poor results in the last model, even after the correction of mislabelled requests and artificial oversampling methods, and as such it should be addressed in detail. One recommendation to be given in this area is to merge or remove categories and subcategories that are least populated from the classification model, either to fully consolidating them into more general labels or to assume that for automatic classification they should not be classified of classified as "SB6" for example. Another approach could be to determine the confidence of each prediction that is achieved by the model and if it is under a certain percentage the specific label would be automatically sent for manual classification and as such reducing the impact of a misclassified label.

The SVM algorithms employed to the available corpus has proven to be the more reliable algorithm and it had in general the best results, either when using as vectors applied to the requests subject or the body text, or through the various vectorization methods applied on the body text experiments. BERT also demonstrated to be a viable model for this dataset, since with minimal configuration it consistently was the second-best model, with results close to the SVM model, as such we would recommend further finetuning of this model since it would likely improve its results and have a chance to outperform SVM. The downside of this approach: the BERT model is more computationally expensive.

The hierarchical model demonstrated to be a viable option to deal with the unbalanced dataset. Whilst using the results from SVM Group classification, with the requests subject, that had the F-Score of 94.36%, it had the score of 93.41%. This result proved that with focused models for each group, and with a smaller feature pool the results were promising. This model when intertwined with SVM for Group classification and BERT for subsequent Category and Subcategory classification did also produce the best results

for Sub categorical classification with the score of 76.95%, when compared to the second-best score of 70.90%. A model could be experimented where SVM as independents model would classify the Category and BERT would classify the subcategory.

Another set of experiments that could have been tried would be implying the same experiments done on the second set of models where the hypermeters where automatically set and applying the base vectorization to verify if the subject could achieve better. Yet with the results achieved it demonstrated that regarding this sector the request subject is also a viable option for development. This may be due the subject is a more concise and reduce set of information, when compared to the request's body text, and produced less noise training the models.

Chapter 7 – Questionnaire

To gauge the acceptance level the methods developed would have on the current helpdesk team a brief quantitative and longitudinal questionnaire was launched. This questionnaire consisted of four brief questions that would be answered with the values on a scale from one to ten.

Since this model was a first approach of such technology the questionnaire was repeated three times, in a longitudinal fashion. Once before any model was executed to create a baseline of the team's opinion on this subject. A second questionnaire was launched when the first model had developed results and a third was launched when the final model was concluded. To maintain a logical continuity on the line of inquiry, this questionnaire was the same for each launch. However, each release would have a different introductory text:

- 1st launch, described the project purpose and how the model would be executed. This would give the inquired element a brief description of the project and where it would be used. This questionnaire was made available at 17/01/2023 and had 14 respondents.
- 2nd launch, the introductory text would reveal the results obtained at that stage and give a confusion matrix of the same results so the user could verify where the model was failing. Each specific technical term or image was briefly explained as to not overwhelm the respondent. To be noted that this model used the requests subject to classify the request group and category. This questionnaire was made available at 25/05/2023 and had 10 respondents.
- 3rd launch, the user was informed of the results, using the same philosophy as described above, and added a brief comparison against the previous model. The second model used the request body text to classify the request's group, category, and subcategory. This questionnaire was launched at 31/08/2023 and had 8 respondents.

For this questionnaire the study population consisted of the elements of the Port Authority helpdesk team, that consisted of 16 elements with ages ranging from 19 to 60 and having different levels of education ranging from high school diploma to master's

degree. To be noted that during the launches not all elements responded for various motives.

The questions were divided into four main inquiry areas, that we will delve into more detail bellow:

Q1. From 1 to 10 (were 1 means totally unsatisfied and 10 totally satisfied) how do you gauge the current helpdesk request distribution?

- This question intended to assess the satisfaction level regarding the helpdesk request distribution at the time. It sought to quantify the respondent's perception of how effectively these requests are distributed among the different groups, categories, and subcategories. By using a numerical scale ranging from 1 to 10 we intent to gather quantitative data regarding this satisfaction levels.

Q2. From 1 to 10 (were 1 means very irrelevant and 10 very relevant) how relevant you believe that Helpdesk request distribution is.

- This line on inquiry sought to quantify the importance placed by the respondents on helpdesk request distribution. By quantifying this perceived importance, we can better gauge the level of quality that is expected from the model, and also better perceive how this activity impacts on the day-to-day operations.

Q3. From 1 to 10 (were 1 means very irrelevant and 10 very relevant) on a broader scope how do you classify the relevance of process automatization?

- This inquiry intended to understand the importance of automation of processes on a broader context. By quantifying this perceived importance within the Helpdesk team, we can gauge how relevant they believe this subject and if they believe that the automation of processes is a relevant approach.

Q4. From 1 to 10 (were 1 means not confident at all and 10 totally confident) how to confident are you that the current manual Helpdesk Request could be automated, using machine learning algorithms?

- This question intended to quantify the respondent's perceptions regarding the feasibility and effectiveness of a machine learning algorithm for the

task of classifying helpdesk requests. By quantifying this perception, we can gauge the willingness of the stake holders involved to embrace this technology or concerns that must be addressed during the implementation of the model into a prototype.

In the table 15 we will delve on the average results given on each release by question, maintaining the same numeration as above described.

Table 15 – Questionnaire average response results.

Question	1st Release	2nd Release	3rd Release	Average
Q1	8	8	8.4	8.1
Q2	9.1	9.0	9.6	9.2
Q3	7.8	8.1	6.1	7.5
Q4	7.9	8.6	7.0	7.9

When enquired about the current helpdesk request distribution the respondents, in terms of the first question, there was a stable high opinion on the current manual distribution of Helpdesk Request distribution. This is quantified by an average score of 8, 8 and 8.4 on the first, second and third release of the questionnaire release. In general these results mean that the current team has a positive overview of the current method, that may be due to the situation that the current distribution is made by two elements or a perceived that the requests have been correctly distributed or its misclassification didn't incur an individual constraint, yet this claims would need deeper investigation or inquiry in order to gather further information on this matter.

On the second question, where the perceived relevance of helpdesk requests distribution is gauged, the results were high and, on any release, went under a quantitative average of 9. Importantly the last result showed a median average of 9.6. These results demonstrated that the respondents, consisting of the elements of the helpdesk team, consistently attribute a very high relevance to the request distribution in their daily operations.

Regarding the results of the third question, we can observe the largest variance between releases. Having an initial nominal average of 7.8 suggesting a moderate

perceived relevance into a slight increase into 8.1 on the second release, yet, on the third release there was a major drop to 6.8. These results show that along the time the perception on the automatization of processes had a high level of volatility. That may have been due to internal factors such as implementation other technologies or enhancement on the current existing processes, or due to external factors, such has a large release of news or technologies regarding this kind of technology. Yet further analysis and investigation should be developed to further investigate and address this volatile perception, and to qualm the concerns and considerations of the helpdesk team members.

The results for question 4 initially the responds had a high confidence level regarding the potential capability to automatizes the helpdesk requests classification, with a nominal average of 7.9. This confidence was reinforced in the second questionnaire release where the nominal average was 8.6. This confidence was not reciprocated on the last release where there was a major drop in confidence that translated into a nominal average of 7.0. These results demonstrated that the respondents started with high expectations regarding the automation of this process and presentation of the results of the first model, with high good results for group and category classification, further improved this perception. However, exposure to the results of the last model, which demonstrated a lack of a major improvement, in a numerical sense regarding the group and category classification and a subcategory classification below 80% may have played a significant role in the loss of confidence on the automatization of this process. Yet to further understand this trend further investigation would be required.

In general, the results suggest that the respondents, helpdesk team, consider that the helpdesk request distribution is highly relevant to their operations and have a high and stable opinion regarding the current distribution. Regarding the general automatization process they are bit more on the fence, and their opinion fluctuates along the questionnaire releases, yet they have demonstrated a relatively high confidence in the automatization of this activity. The drop in confidence, as observed on the third question, has also been noticed on the question regarding the Helpdesk request automation, even though the third results and the average show that they have quite a positive view on this matter. To further develop this line of inquiry a deeper understanding would be necessary and could be achieved by another qualitative analysis, through new questionnaire developed for the respondents on the current questionnaire. Another approach could be to investigate other projects where a similar questionnaire has been executed to compare if the results

presented here are exclusive to this sample or if they reflect a general trend on this sector/process. A longitudinal study could also be carried out with a view to ascertaining how the team would react to a prototype implementation of this model, in order to study model/prototype release perception and realising this questionnaire on a fixed time schedule in order to see if these responses are affected by the models results or by the externals factors.

Chapter 8 – Conclusion

Considering only the requests subject, our study shows that the SVM model for Group classification, and the SVM in a hierarchical architecture for the Category classification have produced the best results with a F1-Score 94.36% and 93.41% respectively and a score of 92.48% when considering the category classification in the non-hierarchical model with SVM. It should be noted that when these two models are combined and data preprocessing was added, the total simulation time for this system was about 10 minutes, meaning that it gives good results with low resource allocation.

While considering the executed experiments with the non-hierarchical models using the request's body text for vectorization, we were able to achieve F1-Scores of 95.42% and 93.47% that when compared with the request's subject experiments in a non-hierarchical model have in general a 1% improvement. To be noted that on this model we were able to further develop the experiments in hyperparameter configuration has suggest on [31] and have also seen that there was a slight advantage with the large Corpus, though at the cost of increased computational strain on the available equipment for this experiments, that only translated on a 0.06% improvement over the category classification with request's subject vectorization. On these experiments we have also added the classification of the subject, obtaining a F1-Score of 70.90% where we can conclude that the applied methods weren't enough to deal with the unbalanced dataset, new methods will be necessary has well a redesign of the subcategory model if it is to be included in the prototype.

The simulation results of the hierarchical models for Group and Category classification with F1-Score of 95.42% and 93.47%, respectively, demonstrated that for this task these models are a viable solution. Yet, when regarding the subcategory classification the best achieved F1-Score of 77.23% also shows that further development would be necessary to improve the model.

As identified in [31] another promising option is the BERT models, since they have good results, but these models most have a deeper fine-tuning since they performed well even though further tunning has been left out on these experiments. There are also new models being released that could have an interesting result such has GPT, LLAMA and SAMBA. The major drawback of these models is the resource consumption necessary for

their training, even when considering that the Corpora is quite limited, when compared with their original data source.

As a result of these experiments, we can conclude that the models for Group and Category have been quite successful and are a good option for the task at hand and can be a starting point for developing a smart response Helpdesk system for the Port Authority Community, using tools belonging to the generative AI field. Yet the experiments with the subject and body text have demonstrated that even a smaller helpdesk team with limited resources, can developed their own model using the requests subject to classify the requests that is quite simpler to handle, in terms of factors to consider on the vectorization, also being cost effective in computational means.

When gauging the user's interaction through the questionnaire, the Helpdesk users perceived as highly relevant the ticket classification and its accuracy and understand its added value. They also perceived process automation positively, even if there was a significant reduction in the last questionnaire release. Regarding the ability to automatize the classification of helpdesk requests they also have a positive belief that this process can be accomplished. Yet these results have limited statistical significance, meaning that for deeper comprehension a more thorough auscultation should be developed.

References

[1] PINTO, Cláudio José. Impactos organizacionais, informacionais e tecnológicos da implementação da Diretiva 2010/65/UE: uma proposta de solução nacional. 2016. PhD Thesis. Instituto Politécnico de Setúbal. Escola Superior de Ciências Empresariais.

[2] DOS SANTOS, Diogo Guilherme Marques. Serviço de Helpdesk Automático. 2020. PhD Thesis. Instituto Politécnico do Porto (Portugal).

[3] JÄNTTI, Marko; CATER-STEEL, Aileen; SHRESTHA, Anup. Towards an improved it service desk system and processes: a case study. International Journal on Advances in Systems and Measurements, 2012, 5.3 & 4: 203-215.

[4] AGARWAL, Shivali; SINDHGATTA, Renuka; SENGUPTA, Bikram. SmartDispatch: enabling efficient ticket dispatch in an IT service environment. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 2012. p. 1393-1401.

[5] MANDAL, Atri, et al. Cognitive system to achieve human-level accuracy in automated assignment of helpdesk email tickets. In: International Conference on Service-Oriented Computing. Springer, Cham, 2018. p. 332-341.

[6] REVINA, Aleksandra; BUZA, Krisztian; MEISTER, Vera G. IT ticket classification: the simpler, the better. IEEE Access, 2020, 8: 193380-193395.

[7] XU, Jian, et al. Signature based trouble ticket classification. Future Generation Computer Systems, 2018, 78: 41-58.

[8] ZICARI, P., et al. Combining deep ensemble learning and explanation for intelligent ticket management. Expert Systems with Applications, 2022, 206: 117815.

[9] KALLIS, Rafael, et al. Ticket tagger: Machine learning driven issue classification. In: 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2019. p. 406-409.

[10] KHOWONGPRASOED, Kraidet; TITIJAROONROJ, Taravichet. Automatic Thai Ticket Classification By Using Machine Learning For IT Infrastructure Company. In: 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE). IEEE, 2022. p. 1-6.

[11] AL-HAWARI, Feras; BARHAM, Hala. A machine learning based help desk system for IT service management. Journal of King Saud University-Computer and Information Sciences, 2021, 33.6: 702-718.

[12] LARASATI, P. D., et al. Chatbot helpdesk design for digital customer service. Applied Engineering and Technology, 2022, 1.1.

[13] PARAMESH, S. P.; SHREEDHARA, K. S. Automated IT service desk systems using machine learning techniques. In: Data Analytics and Learning. Springer, Singapore, 2019. p. 331-346.

[14] NEVES, Vilma Carina Simões, RIBEIRO, Ricardo Daniel Faro Marques Ribeiro, MAMEDE Nuno João Neves. Automatic Classification of Correspondence from a Public Institution. 2021. Masters Thesis.

[15] FAZENDEIRO, André Miguel Balau, RIBEIRO, Ricardo Daniel Faro Marques Ribeiro, MAMEDE Nuno João Neves. Automatic Correspondence Distribution for a Public Institution. 2021. Masters Thesis.

[16] LARASATI, P. D., et al. Chatbot helpdesk design for digital customer service. Applied Engineering and Technology, 2022, 1.1.

[17] CARDOSO, Pedro Nuno de Sousa. Automatização de Processos Helpdesk, Utilizando Recursos de Inteligência Artificial. 2019. PhD Thesis.

[18] MAROM, Yuval; ZUKERMAN, Ingrid. An empirical study of corpus-based response automation methods for an e-mail-based help-desk domain. Computational Linguistics, 2009, 35.4: 597-635.

[19] MANDAL, Atri, et al. Cognitive system to achieve human-level accuracy in automated assignment of helpdesk email tickets. In: International Conference on Service-Oriented Computing. Springer, Cham, 2018. p. 332-341.

[20] AL-HAWARI, Feras; BARHAM, Hala. A machine learning based help desk system for IT service management. Journal of King Saud University-Computer and Information Sciences, 2021, 33.6: 702-718.

[21] MARCUZZO, Matteo, et al. A multi-level approach for hierarchical Ticket Classification. In: Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022). 2022. p. 201-214.

[22] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems, 2017, 30.

[23] HACOHEN-KERNER, Yaakov; MILLER, Daniel; YIGAL, Yair. The influence of preprocessing on text classification using a bag-of-words representation. PloS one, 2020, 15.5: e0232525.

[24] JOACHIMS, Thorsten. Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. Springer, Berlin, Heidelberg, 1998. p. 137-142.

[25] GUDIVADA, Venkat N., et al. Cognitive analytics: Going beyond big data analytics and machine learning. In: Handbook of statistics. Elsevier, 2016. p. 169-205.

[26] CHARBUTY, Bahzad; ABDULAZEEZ, Adnan. Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2021, 2.01: 20-28.

[27] ZAFARI, Mostafa, et al. A practical model for the evaluation of high school student performance based on machine learning. Applied Sciences, 2021, 11.23: 11534.

[28] SUNDERMEYER, Martin; SCHLÜTER, Ralf; NEY, Hermann. LSTM neural networks for language modelling. In: Thirteenth annual conference of the international speech communication association. 2012.

[29] ÖZÇIFT, Akın, et al. Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 2021, 62.2: 226-238.

[30] SEGURA-BEDMAR, Isabel; CAMINO-PERDONES, David; GUERRERO-ASPIZUA, Sara. Exploring deep learning methods for recognizing rare diseases and their clinical manifestations from texts. BMC bioinformatics, 2022, 23.1: 263.

[31] MARTINS, Samuel António Beecher; GARRIDO, Nuno; SEBASTIÃO, Pedro. Port request classification automation through NLP. Procedia Computer Science, 2024, 239: 1927-1934.

Appendix A

Complete configuration for each Body Text vectorization experiment while classifying Request Group

	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	'C': 10, 'gamma': 'scale', 'kernel': 'linear'	C=1.0, kernel='linear', degree=3, gamma='auto'	C': 10, 'gamma': 'scale', 'kernel': 'linear'	C': 10, 'gamma': 'scale', 'kernel': 'linear'	C=1.0, kernel='linear', degree=3, gamma='auto'
Random Forest	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'
Decision Three	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200
LSTM	5 epocs	7 epocs	8 epocs	6 epocs	31 epocs

Appendix B

Complete F1-Score results for each Body Text vectorization experiment while classifying Request Group

Result	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	0.9197	0.9229	0.9198	0.9219	0.9347
Random Forest	0.8745	0.8846	0.8722	0.9024	0.8877
Decision Three	0.8734	0.8908	0.8739	0.8734	0.8912
LSTM	0.8708	0.8883	0.8667	0.8796	0.8442
BERT	N/A	N/A	N/A	N/A	0.9233

Appendix C

Complete configuration for each Body Text vectorization experiment while classifying Request Category

	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	'C': 1, 'gamma': 'scale', 'kernel': 'linear'	'C': 10, 'gamma': 'scale', 'kernel': 'rbf'	'C': 10, 'gamma': 'scale', 'kernel': 'linear'	'C': 1, 'gamma': 'scale', 'kernel': 'linear'	'C': 10, 'gamma': 'scale', 'kernel': 'linear'
Random Forest	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300
Decision Three	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'
LSTM	5 epocs	6 epocs	5 epocs	4 epocs	37 epocs

Appendix D

Complete F1-Score results for each Body Text vectorization experiment while classifying Request Category

Result	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	0.9427	0.9461	0.9436	0.9462	0.9542
Random Forest	0.9064	0.9192	0.9011	0.9278	0.9238
Decision Three	0.9005	0.9138	0.8861	0.9137	0.9276
LSTM	0.9068	0.9172	0.9080	0.9177	0.9065
BERT	N/A	N/A	N/A	N/A	0.9487

Appendix E

Complete configuration for each Body Text vectorization experiment while classifying Request Subcategory

	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	C=1.0, kernel='linear', degree=3, gamma='auto'	'C': 10, 'gamma': 'scale', 'kernel': 'linear'	C': 10, 'gamma': 'scale', 'kernel': 'linear'	C=1.0, kernel='linear', degree=3, gamma='auto'	'C': 10, 'gamma': 'scale', 'kernel': 'linear'
Random Forest	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 300
Decision Three	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'	criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'
LSTM	13 epocs	14 epocs	18 epocs	8 epocs	14 epocs

Appendix F

Complete F1-Score results for each Body Text vectorization experiment while classifying Request Subcategory

Result	Complete	Lemmatization	Pos Tag	Tokenized	Base
SVM	0.6908	0.6869	0.6862	0.6904	0.7090
Random Forest	0.6491	0.6616	0.6517	0.6646	0.7031
Decision Three	0.6289	0.6276	0.6240	0.6377	0.6571
LSTM	0.5785	0.5941	0.5808	0.6078	0.4973
BERT	N/A	N/A	N/A	N/A	0.7031