

Automated Data Classification And Pre-Processing Of It Service Tickets For Future Machine Learning Predictions

Faraz Islam¹, Tasneem Ahmed^{*2}

^{1,2} Department of Computer Application, Integral University, Lucknow, India-226026,

^{*}Corresponding Author: Tasneem Ahmed^{*}

Abstract: Nowadays, many IT help desks depend on ticket systems when dealing with issues, inquiries, or system failures. Working through these manually often drags the pace, consumes extra work hours, and puts schedules under pressure. Although operational platforms gather vast amounts of textual records, those rarely fit algorithmic needs without adjustment. A lot comes cluttered - filled with irregularities that block smooth analysis. Consider missing pieces of information, duplicated entries, inconsistent formats, embedded snippets of code, misplaced tags, multilingual phrases, or extended freeform descriptions tough to interpret by machines. One way to study raw IT support tickets is by tracking their path across an entire system, beginning at ManageEngine ServiceDesk Plus, where they are made. From there, they travel toward processing inside a private SQL Server environment. Step follows step using R-based utilities that tidy and reshape textual data ahead of further analysis. Early on, detection shifts focus to non-English texts, and these get set aside quickly. Following that stage, duplicates are removed, leaving behind only distinct words ready for a closer look. Once filtered, words without clear value disappear. Then comes reducing terms to basic versions, so "running" becomes "run", evening out small mismatches. Strange marks or uneven layout get removed. Every line shift into a consistent format, and later on, the key terms become clearer once the focus shifts towards the most repeated words. Visual maps of frequent word pairings uncover patterns not obvious at first glance. Meanwhile, broader examinations help keep rare outliers from skewing interpretation. From a total of 33,059 service tickets, every data entry contains structured fields alongside text columns such as Subject and Description. The results part shows that rigorous data cleansing improves the data consistency, produces a high-quality dataset, and reduces noise, which is suitable for training datasets using a machine learning algorithm. For future work, the focus will be on integrating the cleaned dataset with Azure Machine Learning Studio for automated ticket dispatching. Also, aiming at the optimization of the proposed model through hyperparameter tuning, continuous retraining with live datasets, and real-time deployment to enhance the service efficiency, SLA compliance, and overall satisfaction of the users.

Keywords: IT Service Management (ITSM), Ticket Classification, Text Pre- processing, Data Cleansing, ManageEngine ServiceDesk Plus (MSP).

I. Introduction

In the modern digital ecosystem, companies accumulate large amounts of unstructured text, especially on the IT support websites of the company [1, 2]. The records left by users (help tickets, summaries of an issue, reports of an event, and messages between each other) bring together large and extensive data sets; when carefully handled, these data sets provide valuable insights into patterns and avenues of automated solutions. However, as the information is not touched or corrupted, it becomes useless. Before analyzing or applying machine-learning methods, data cleansing becomes critically important, not optional, beforehand. The raw data generated by instruments implemented in controlled services are often full of errors: spelling mistakes coexist with malfunctioning designs, duplicate records appear, unnecessary tags remain, and capitalization is unpredictable and unsteady in nature [3]. Normally ignored, ordinary words like are, the or her are common in what users provide, and so are punctuations and numbers, which can hardly be used in the analysis to provide semantics to the text at all [4]. Left unchecked, these unprocessed inputs have a possibility of biasing results in activities like clustering, labeling, or intent detection [5]. This paper is based on the

lifecycle of textual data in total, such as its extraction, refinement, and preparation to be used in training models to be automated. With examples of real records of IT helpdesk, this work outlines over ten crucial cleaning operations. They are deleting redundant entries, deleting unused fields, filtering the repetitive or uncommon words, converting verbs to their stem forms, deleting embedded code markers, removing the non-letter characters, and standardizing the cases of all characters [6]. One of the central reasons why machine learning fails in non-controlled laboratory conditions is associated with poor text preparation [7]. Even though the lion's share of research is often given to algorithmic development, the quality of raw data is often given second-best consideration. In the context of natural language processing, such an imbalance is often vividly expressed by the situation in which valuable information is often lost in the noise of the background noise [5]. Lack of stringent cleaning protocols leads to poor pattern identification, causing distorted outputs and poor handling of new cases.

Fundamentally, this study highlights the importance of accurate data refinement as the underlying platform to automatically route IT tickets. Processed textual information makes it easier to perform better tokenization, count frequencies, and separate representations of features to use in training models [8]. It has been shown that the deletion of digits attached to letters, the unnecessary whitespace, and the replacement of Latin-1 symbols with simple ASCII characters can make the entries more consistent. Omission of non-standard stop phrases, e.g., names of staff, names of systems or teams, etc., eliminates factors that do not give any analytical value. Having removed such distractions, algorithms may be used to pinpoint meaningful content in contrast to focusing on the insignificant repetitions [4]. The visual analyses that are used through word clouds help in discovering any rare or dominant terms; elimination of the word clouds is known to sharpen the corpus and make it express more helpful and dense signals. The resulting dataset is therefore narrower with signal predominating noise. Moreover, the stemming procedures and case normalization decrease the number of distinct tokens, thus easing the process of pattern finding. In the process of stemming, morphological variants, like running, are reduced to a canonical form, such as run, which allows word variants to be more readily handled in a consistent way [9]. Lowercasing all text makes the cases of different instances of the same word similar and thus eliminates duplication during feature construction. One particular problem investigated in this paper concerns the elimination of latent XML and HTML tags that are common in ITSM export, which add visual noise to the analysis and block it until it is eliminated [6].

Treatment of irregular text structures represents the other aspect of this work, which involves content coded in Latin-1 characters or phrases joined together by omitted spaces. By fixing such anomalies, tokenization and parsing can be performed effectively, which are necessary conditions before further NLP processes can be performed. An important cleaning process involves the elimination of quotations, unnecessary whitespace, and sorting out of numbers that have no meaning to the text [7]. These features are minor in scale but can disproportionately bias the distribution of word frequencies and change the relationship between terms. Getting rid of alphanumeric bits like error404 helps to narrow down to words that have significance in the dataset. On the whole, this study shows the relevance of the preprocessing process in converting raw ITSM tickets into machine-readable ones [10]. The findings show that the text preprocessing of quality can directly influence the functionality of classification systems when accomplished as a part of automated workflows. It is not an easy list of cleaning operations, as the approach offers general structures that could be further developed in terms of thorough research of the real-time text sorting and smart support ticket management, where such a tool as Azure ML and R-based models could be used in particular.

II. Literature Review

The studies are unanimous about the importance of text data cleaning in terms of dealing with NLP and ML systems. Preparation of input is not a small matter; it dictates what models will be able to learn - the results of such works as Ghosh et al. [11] are resolute. Han et al. [12] solidify this conception, proving that a practice of quality of raw text literally directly changes the results in the domains of heavy written traffic, e.g., tech support desks. For decades, thousands of papers have studied the ways of organizing unstructured language inputs. The first step for some is to get rid of common filler words; the second is to restructure the verbs by stem cutting or by mapping out roots. Replacing letters with similar ones frequently occurs together with the erasing of the letters that are no longer required. Manning, together with Schütze [13], as these steps reduce junk and redundant sections, patterns of words are simplified, besides decreasing processing in computers. Recently, the focus has shifted to the issue of the removal of stop words in relation to particular domains. General lists generated by such tools as NLTK or spaCy often have limited application when used with corporate data. Rather, specific collections - collections that screen out things such as employee codes, automated messages, or the re-use of administrative wording- are more likely to increase accuracy in the company setting, as results like those of Parameswaran et al. suggest [14]. In the case of clean-up services related to IT services, it can be seen that when it comes to cleaning service management, the cleaning service request records are seldom penetrated. Nonetheless, such attempts were made by Rahman et al. [15] and Joshi et al. [16] have presented ways of classifying helpdesk tickets with the use of simple natural language processing procedures. Some of these studies are incomplete in terms of exhaustive ways of cleaning data, where usually textual inputs are assumed to be available to be utilized. Due to this assumption, the project pays attention to a complete chain of actions that are required to prepare ITSM tickets and make them useful for the implementation of machine learning applications. Other studies consider visuals such as word clouds when performing preliminary data cleanup. Kumar et al. [17], before training models, identify the words that are rarely read or are messy, and can be made easier by pointing out these displays. Cutting odd words is quicker since they appear clearly. Although their primary task is exploration, feature sharpening is performed step by step, adjusting to them. Ready-made NLP solutions are encountered frequently in previous practice, but scholars believe that custom-made solutions better suit the business purpose. Custom flows are more important in cases when the real-world systems require accuracy [18, 19, 20]. The papers that have been written about earlier support this requirement based on industry examples. The structured tools, in particular MSP, where the information may include HTML/XML tags, coded symbols, or unusual layouts, are especially applicable [21]. Even though there is little previous research regarding the means of dealing with these quirks, this study addresses them directly.

In retrospect, advances in deep learning and transformers did not reduce the necessity of well-structured and neat data. Even dynamic models such as BERT and GPT cannot work without sound input to operate. By passing by raw information without first going through steam cleaning techniques, models can cling onto noise, making a distorted rendition of broken phrases and then providing poor results [22]. There is one interesting result - preprocessing influences results more than it would. Recent work by Kumar et al. [23] demonstrates the reduction of power requirement in IoT systems through smart data handling. The technique they use is meticulous refinement, and then they start modeling. Efficaciousness is not only the byproduct of the algorithms, but also predilection in design. The real-world tests showed apparent improvement when adjusted during the feature extraction. Using transformer architectures like BERT and RoBERTa, Ahmad, along with

Kumar [24, 25], classified emotions and sentiments in tweets about the COVID-19 crisis, showing how careful cleaning and splitting of text improves outcomes.

Instead of standard modeling, Tripathi et al. [26] built a machine learning system for predicting diabetes from data gathered by wearable IoT sensors, where balancing skewed classes via SMOTE played a key role alongside tuning settings. In another work, forecasting health patterns after pandemics became the focus for Tripathi together with Pathak [27], who demonstrated that shaping inputs through preprocessing and crafting meaningful variables boost forecast precision. What ties these works together is clear without thoughtful preparation of data and attention to features relevant to the field; even advanced models may underperform. Despite growing interest, few studies dig into how cleaning methods shape outcomes for IT service texts. What stands out? Strong results depend heavily on careful, situation-specific preparation of data. The majority of the current initiatives bypass them or complete them hastily. This research paper addresses the gap by providing a stepwise process that can be adopted by anyone. The approach introduces coherence where it did not exist before, with arbitrariness dominating as the main characteristic of ITSM inputs [28]. Advancements in this regard quietly leave better classifiers in the future.

III. Method

The approach that is used in this research entails the use of a systematic pipeline that transforms raw information published by the IT service request into a clean and analyzable format that can be utilized in text mining and machine learning processes. Developed on the basis of an operational IT service management platform, the data contains both structured properties and unstructured textual fields, which means that it requires careful preparation. This makes each subsection of the Method chapter clearer about the context of the system, specifying the nature of the dataset used and specifying the procedures used to preprocess it. Service Request System Overview outlines the processes through which tickets are created, stored, and managed in the MSP platform and its SQL server-based infrastructure. Having an overall view of the process of work, the semantics of different fields, including Subject, Description, and Category, provides the necessary background for understanding ticket trends and creating a useful pre-processing plan.

The Data Description section is a systematized restatement of the form and composition of the extracted dataset, listing the count of observations, variables of interest, and the types of data. This summary highlights the importance of text-based areas and points out the possible pitfalls, that is, noise, inconsistencies, and heterogeneous data formats, which need to be addressed before analysis. The Data Pre-processing section provides the statement of the transformation pipeline, which causes the textual content to become vulnerable to modeling. It also has language identification to enlist only the tickets that include only English, it also cleans the tickets to eliminate the noise and normalization of the orthography, and it also extracts the key words to eliminate the substantial words and retain the ones that were used in the Subject and Description fields. These steps are separately presented in subsections such as Word Cloud, Language Detection, Filtering, and Cleaning, which ascribe to them the significance that it pays to data quality. Finally, the Exploratory Analysis provides an initial analysis of the number of terms, the number of categories, and trends of emergent patterns of the curated data. These are insights based on which appropriate modeling techniques are selected, as well as building the robust predictive systems of automated ticket classification and routing. Combined, such a methodological framework would ensure that the dataset was conceptualized, enriched in analysis, and cleansed in a systematic manner, hence allowing the natural language

processing and machine learning techniques to be successfully applied to the field of IT service management.

III.I. Service Request System Overview

XYZ GmbH has a Managed Service Provider (MSP) to manage its IT services, and the data is kept in a database that is hosted on Microsoft SQL Server on the on premises. The front-end interface of the MSP is used to record ticket requests sent by employees and clients to it; they are then stored in the MSP database. SQL Server cluster has failover features, which also increase the availability and reliability of the system in general. The shared storage architecture adopted by the organization brings about a single point of failure. Service Requests have the following attributes: CreatedTime, Subject, Description, category, and subcategory. Service Requests have resolution priorities allocated based on the Service Level Agreement (SLA) as shown in Table 1. Cases where the quality of a service delivered by IT is reduced or where the unexpected interruption or failure of some configuration item precedes the negative impact on the service are considered incidents. That is, when an IT service has stopped, or its quality or functionality is diminished, or it malfunctions, it will be considered an incident [29].

Tables 1 and 2, respectively, show the priority matrices of Service Requests and Incidents, hence giving the business context under which the trends in the tickets are to be interpreted and the predictive models can be developed.

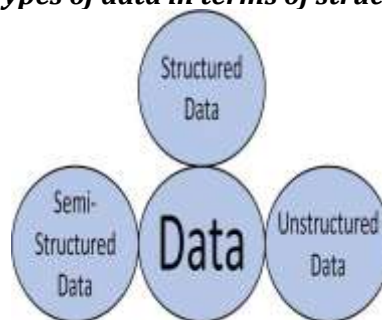
Table 1: Service Request Priority Matrix.

Priority	Resolution Time	Applicable Hours	SLR
Normal	3 DB	Business Hours	95%
Urgent	Asap	Business Hours	95%

Table 2: Incident Priority Matrix.

Priority	Resp. Time	Res. Time	Hours	SLR	Report
Critical	15 min	4H	24x7	95%	10-20 DB
High	2 BH	8 BH	BH	95%	20-30 BD
Medium	4 BH	2 BD	BH	95%	No Request
Low	8 BH	3 BD	BH	95%	No Request

Figure 1: Types of data in terms of structure.



III.II. Data Characteristics and Structure

Determining the characteristics of data is a critical step in the process of moving on towards text analysis, since it determines the choice of the right data pre- processing methods and largely enhances the outcome gained. It is important to know the nature of data structures in which the research dataset is structured before discussing the data characteristics of the research data. Figure 1 depicts the nature of data structures arranged in which the research data is arranged. Having analyzed the situation and considered Table 7, the character description it provides, one might easily make out the organization of the data used to analyze the text. The investigation employs MSP to receive data from end users; the data are stored in structured information in a database. These types of databases share well-defined attributes and respect the hierarchical relationship existing between fields and records.

The overall knowledge of the type of data contained in this structured dataset is essential in the process of data pre-processing. A data type is a property of a coordinate that defines the type of information it can hold, such as an integer, a date and time stamp, a character string, an array, or an alphanumeric number. SQL Server, a column, variable, and parameter are the values that are related to a particular data type [30]. Tables 3 and 4, as well as Tables 5 and 6, in the appendix outline the design of each column in the MSP database, including numeric, date and time, and character columns. These types are fundamental in understanding the techniques to be used on pre-processing, like date conversion to analyze the trends, coding of categorical variables, and sanitization of text fields.

Table 3: Numeric Data Types.

Type	Lower Limit	Upper Limit	Memory
bigint	$-2^{63} (-9, 223, 372, 854, 775, 808)$	$2^{63} - 1 (-9, 223, 372, 854, 775, 807)$	8 bytes
int	$-2^{31} (-2, 147, 483, 648)$	$2^{31} - 1 (-2, 147, 483, 647)$	4 bytes

Table 4: Approximate Numeric Data Types.

Type	Lower Limit	Upper Limit	Memory	Precision
float(n)	$-1.79E + 308$	$1.79E + 308$	Depends on n	7 Digit
real	$-3.40E + 38$	$3.40E + 38$	4 bytes	15 Digit

Table 5: Date & Time Data Types.

Type	Size	Accuracy	Lower Range	Upper Range
datetime	8 bytes	Rounded to increments of .000, .003, .007	1753/01/01	9999/12/31
date	3 bytes	1 day	0001/01/01	9999/12/31
time	5 bytes	100 nanoseconds	00:00:00.000000	23:59:59.999999

Table 6: Character Strings Data Types.

Type	Size	Accuracy	Lower Range	Upper Range
char(n)	n bytes	Fixed-length string	1 chars	8,000 chars
varchar(n)	$n + 2$ bytes	Variable-length string	0 chars	8,000 chars
nchar(n)	$2n$ bytes	Fixed-length string	1 chars	4,000 chars
nvarchar(n)	$2n + 2$ bytes	Variable-length string	0 chars	4,000 chars

III.III. Data Description

In XYZ GmbH, the MSP manages the ticketing system and stores the data on a database that runs on a Microsoft SQL Server located on premises. The front- end interface of the MSP records the ticket requests made by the employee and clients, and then they are stored in the MSP database. SQL Server cluster has failed over mechanisms, hence enhancing cross-over availability and reliability of the system. The shared storage architecture is used by the organization and creates a single point of failure.

The information that is stored in MSP represents time-based data; the appropriate observations are retrieved through joints between numerous tables and combined into a single entity. The primary data that will be used in this study are 33,059 observations and six column-level variables. As shown in Table 7, the columns have various data types, attributes, and value ranges, and present important variables that will be used in analysis, including the use of text mining: e.g., the columns of Subject and Description, and that of WorkorderID as a unique identifier.

Table 7: Description of XYZ GmbH's data structure.

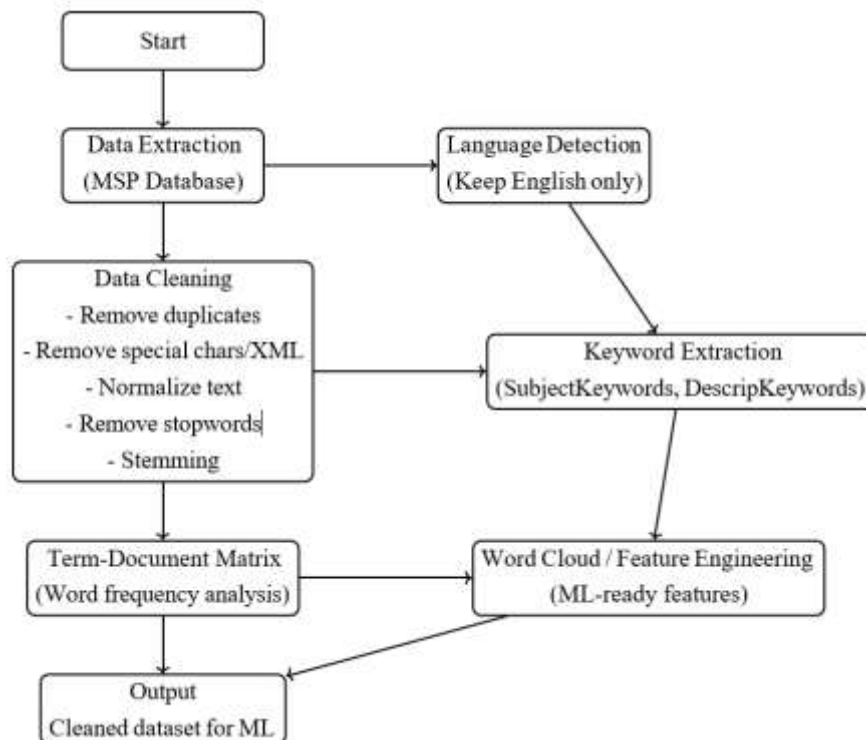
Column	Type	Char	Size	Description
CreatedTime	datetime2	3	7 Bytes	Date and Time.
WorkOrderID	bigint	5	8 Bytes	A unique number.
Category	nvarchar	100	200 Bytes	Unique categories
SubCategory	nvarchar	100	200 Bytes	Sub-categories.
Subject	nvarchar	250	500 Bytes	Brief description
Description	nvarchar	10^8	2 GBytes	Request Reason.

III.IV. Data Preprocessing

Raw information coming out of the system under IT Service Management is in unstructured text and contains duplication of records, inconsistencies, and redundant information. To make this material machine learning and natural language processing-friendly, a systematic preprocessing pipeline was designed. Figure 2 below presents the sequential steps of this pipeline, illustrating how the entire process of converting raw ticket data to a clean, structured dataset ready to train a model takes place. The pipeline

begins with the process of detection and filtering of the language; thereby, it is now left with the English language written tickets. This is to guarantee the uniformity of language throughout the corpus and alleviate the outcome of noise caused by multilingual contents [31]. Thereafter, data purification is done. They involve the deletion of duplicates, extraneous columns, punctuation, special characters, and unnecessary whitespaces, de-XML/HTML tagging, and disassociating numeric tokens attached to words. They use text normalization steps, e.g., lowercase and stemming processes to standardize words, and the dimensionality of features is minimized among features [32].

Figure 2: Preprocessing workflow for IT service tickets.



Subsequently, the extraction of keywords is done on the subject and description fields of all tickets. The stage identifies the most semantically important words and eliminates frequent stopwords, internal codes, and tokens of lesser frequency. The obtained keywords are captured on specific columns, which provide a dataset that is rich in features to be utilized in downstream analysis [33]. In general, the preprocessing workflow increases the quality of data, eliminates noise, and improves the level of signal-to-noise output, which contributes to building the appropriate predictive models. It has been empirically indicated that similar preprocessing steps can significantly enhance classification in datasets of IT services [34].

III.IV.i. Word Cloud

Word clouds are a type of text-analysis method that reveals the most common words in a piece of text, and thus, gives a graphical display of the data [35]. A word cloud is relatively simple to construct as long as the separate procedural steps are followed. It is possible to use the textual data exploration and visualization of the salient keywords contained in the dataset with the help of the tm package and the word cloud generator [36] and [37]. The word cloud may be of great use in the data cleansing and text-preprocessing tasks that will be performed later in the experiment. The processes involved in the word cloud will

include word stemming, removal of stop-words, and removal of special characters, and will result in a data frame which will list the words as well as their frequency counts. Very low frequency words will be identified in a different data frame and then cross-linked with the main data frame. Since these rare terms would not have any semantic meaning and will not be useful in future predictive analytics, they will be removed from the corpus. The advantages of the use of the word cloud are:

- It adds clarity, visibility, and simplicity. The most occurring words from the corpus stand out as more informative and appealing in a word cloud.
- They are very convincing and intelligent tools that are very easy to understand and to share with others. They are more informative and attractive than tabular data.

Algorithm 1: Word Cloud Generation from MSP Data Set

Require: Preprocessed text corpus

Ensure: Word cloud visualization of frequent terms

1. Construct a Term–Document Matrix (TDM) from the input corpus.
2. Convert the TDM into a numerical matrix.
3. Compute the frequency of each term by summing over all documents.
4. Sort the terms in descending order based on frequency.
5. Store the sorted terms and corresponding frequencies in a data structure.
6. Initialize a random seed for reproducibility.
7. Open a graphical device with a predefined width and height.
8. Select the top N most frequent terms.
9. Generate a word cloud where term size reflects frequency and colors are assigned randomly.
10. Display the word cloud visualization.

Figure 3: Word cloud of the top 300 words in ticket descriptions.



III.IV.ii. Language Detection and Filtering

To ensure uniformity in the textual analysis, as well as to avoid the occurrence of noise due to the inclusion of languages other than English in the dataset, the dataset was narrowed down to English-language tickets only. This is necessary as natural language processing algorithms, e.g., tokenization, stemming, and feature extraction, are severely sensitive to language-dependent syntax and vocabulary. Ticket inclusion in other

languages may generate incorrect frequencies of tokens, misclassification, and reduce the accuracy of the model. Therefore, the language detection, such as in Algorithm 2, was applied to make sure that the dataset is composed only of English.

Algorithm 2: Language Detection and English Text Filtering

Require: Text dataset containing ticket descriptions

Ensure: Dataset filtered to English-language tickets

1. Install the language detection library.
2. Load the language detection module.
3. Apply language detection to each text record.
4. Assign the detected language label to a new column.
5. Filter the dataset to retain records labelled as English.
6. Store the filtered dataset for preprocessing.

III.IV.iii. Cleaning

The contents of the textual information in the Subject and Description columns are full of extraneous factors, and only the keywords are necessary. As a result, two new columns (SubjectKeywords and DescripKeywords) with unfiltered keywords were developed. This conversion will make the subsequent work of analysis of texts smoother with the help of machine learning algorithms. The procedures involved in cleaning up the research data before the actual study are shown in Algorithm 3.

Algorithm 3: Ticket Data Cleaning and Keyword Extraction

Require: Raw ticket dataset containing structured and textual fields

Ensure: Cleaned dataset with extracted keyword features

1. Remove duplicate ticket records based on the unique request identifier.
2. Select relevant attributes, including category, subcategory, description, and timestamp.
3. Subset the dataset to a fixed number of records for analysis.
4. Rename textual fields for consistency.
5. Remove HTML and XML tags from text fields.
6. Convert Latin1-encoded characters to ASCII format.
7. Normalize whitespace by removing extra spaces.
8. Insert spaces between concatenated lowercase and uppercase words.
9. Remove special characters, underscores, and encoded quotation artifacts.
10. Clean subject text by removing redundant encoded terms.
11. Remove domain-specific and multilingual stopwords from subject keywords.
12. Apply stemming to reduce words to their root form.
13. Remove non-alphanumeric symbols from description keywords.
14. Normalize quotation marks and spacing in description text.
15. Apply stemming to description keywords.
16. Store the cleaned and transformed dataset for downstream analysis.

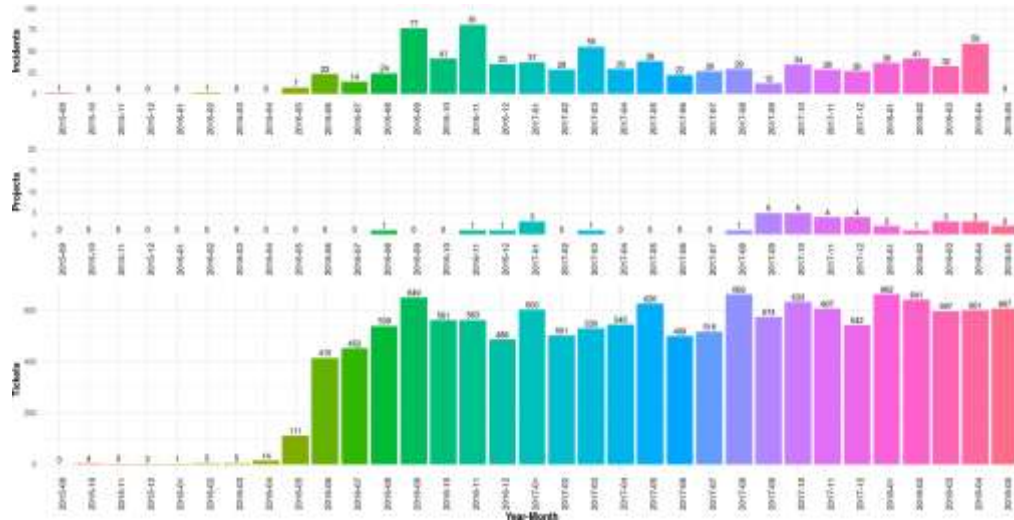
IV. Exploratory Analysis

Figure 4 shows the trend of tickets per time graphically in the form of GGPlot and bar graphs. MSP data shows that the number of tickets takes the leading role in requests with a distinct upward trend over the time frame under analysis, whereas incidents and projects are relatively lower but with spikes suggesting a critical situation or project activity. Outliers and monthly patterns are also conspicuous because of the independent Y-axis scaling and, thus, a more subtle understanding of the historical tendencies is possible. These understandings are not only highlighted. The points of high-ticket volume,

which allow strengthening SLA compliance and resource planning, also provide the premise to further pre-process data and classify the tickets using machine learning, so that the models are trained on informative and representative trends.

It also examined the data set to identify its properties and to outline the possible problems before applying machine learning models to it [38, 7]. Table 8 summarizes the important statistics of the data prior to and after pre-processing, which include the removal of duplicates, language filtering, and cleaning the text.

Figure 4: GGPlot showing the growth of Service Request counts overtime.



As it is shown in Table 8, pre-processing resulted in a significant improvement in the quality of the dataset, which is highly beneficial when it comes to training datasets using Machine Learning techniques [38, 7]. Removal of duplicates, filtering of non-English tickets, removal of stop-words, treatment of special characters, and extraction of keywords enhanced consistency, less noise, and more consistent feature vectors [4, 6]. The last tokens were endowed with significance, and the redundant artefacts were eliminated, whereas the mean text lengths were reduced [33]. Such developments led to the acquisition of a clean dataset that was about to be used in downstream machine learning processes [39, 34].

Table 8: Exploratory Analysis Summary: Ticket Data Before and After Preprocessing.

Feature / Metric	Before Preprocessing	After Preprocessing
Total tickets	33,059	33,059
Duplicates removed	–	1,024
Tickets with non-English text	–	512 removed
Missing Subject / Description	432	0
Average Subject length	12.5	9.2
Average Description length	48.3	39.7
Special characters	High	Removed
Stopwords	Present	Removed
Domain-specific noise	Present	Removed
Unique keywords extracted	N/A	2,350

Top category frequency	HR: 4,210	HR: 4,210
Subcategory uniformity	Skewed	More balanced
Word cloud clarity	Crowded, noisy	Meaningful terms
Token counts per ticket	Highly variable	Reduced variance

IV.I. Results and Discussion

Here, the section defines the outcome of preprocessing and exploratory analysis of the information on IT services tickets and explains its conclusion on the classification provided by the machine learning, as well as the automation of the process of managing the services. The preprocessing pipeline was implemented to convert raw, highly noisy data in the service desk to structured, high-quality data that can be used in predictive modelling. The data obtained through ManageEngine ServiceDesk Plus was 33,059 tickets, and they contained both structured and unstructured data, including subject and description fields. The first inspection showed that there are a few data quality problems: duplication of records, disparate text encoding, the use of XML/HTML tags, bilingual data, and incorrect formatting. These were the causes of higher noise and hindered the results of predictive models. Other works in the recent literature have made similar observations about the negative impact of noisy ITSM data on machine-learning classifiers [40, 39]. Language recognition was used to keep only the English-language tickets, which gave a homogeneous corpus. Removal of duplicate tickets was based on the uniqueness of request identifiers, and irrelevant attributes were to be removed to increase computational efficiency. Text cleaning, which involves removing HTML/XML tags, whitespace normalization, and case normalization, concatenation, word separation, and removal of special characters and numbers, made important contributions to the consistency of text. Word removal (including domain-specific words) that was then enhanced by stemming minimized vocabulary and facilitated term generalization. The extraction of keywords in the subject and description fields created small sets of features, which concentrated on the issue-oriented terms of concern [41, 42].

The word cloud visualization showed that the dominant words were directly related to the service issues, system errors, and the infrastructure components, and low-frequency and generic words were eliminated. An exploratory analysis of the trends in the tickets revealed the constant rise in requests for service, thus indicating the existing operational difficulties. These findings prove that efficient preprocessing boosts the signal-to-noise ratio and creates a solid base to extract features and classify them effectively [43, 44]. Overall, the results confirm that data quality is a key factor in machine-learning success in IT service management. Normalized text is easier to process, tokenize, give better term-frequency distributions, and explain the division between categories, thus making automated classification and routing of tickets possible and accurate with normalized text-reading systems in place [39, 42].

V. Conclusion

Thus, this paper presents a comprehensive system of processing pre-IT service ticket information, thus making it easier to classify with machine learning. The unstructured raw data were transformed into high-quality data of high quality, trained to be read by the machine learning, through systematic cleansing and regularization, and keyword extraction. Primary issues, including duplicate records, inconsistency in formatting, XML/HTML tags, multilingual and noisy artifacts, were appropriately resolved. Very important to the performance of machine-learning classifiers is the quality of the training dataset. This way is the best way to achieve proper and reliable classification by ensuring that there is proper preprocessing of data to achieve less miss-routing and achieve optimal functioning efficiency. The effectiveness of predictive models in the management

of IT services requires thorough knowledge of the data and the use of the relevant preprocessing and visualization procedures to do so. This curated dataset will be combined with Azure ML studio in future work, and continuous training of the models will be performed, along with the hyperparameter optimization and real-time deployment of models in real-world environments using CI/CD pipelines. Further research may evaluate the operational effects of smart ticket automation on user satisfaction, adherence to SLA, and organizational performance. The framework, therefore, gives a good basis for moving away from manual ticket processing to the automated IT service management system.

VI. Funding

The author states that no grant was obtained by funding organizations in the state, commercial, or non-profit sectors to conduct this research. The author financed the study completely.

VII. Conflicts of Interest

There are no conflicts of interest to disclose. The authors make the statement that they do not have any financial, personal, or professional conflicts of interest that may have affected the work reported in this manuscript. The authors also testify that no non-financial conflicting interests, such as political interest, religious interest, ideological interest, academic interest, and intellectual interest, can be viewed as having impacted objectivity or integrity of the research.

VIII. Acknowledgments

The authors are thankful to the Advanced Computing Research Laboratory, Department of Computer Application, Integral University, Lucknow, for providing the necessary support to carry out the research work. The manuscript communication number issued by Integral University is IU/R&D/2025-MCN0005380.

References

- [1] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.
- [2] D. Tikk and Z. T. Kardkova'cs, "Text mining in information systems research: A systematic literature review," *CAIS*, vol. 41, p. Article 12, 2017.
- [3] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–54, 1996.
- [4] C. D. Manning, P. Raghavan, and H. Schu'tze, *Introduction to information retrieval*. Cambridge University Press, 2008.
- [5] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt, "A framework for term weighting approaches in text mining applications," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4243–4251, 2013.
- [6] G. Kaur, I. Saini, and S. S. Chahal, "Text preprocessing for classification using machine learning in information retrieval," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, pp. 738–742, IEEE, 2019.
- [7] M. Rahman and M. Tasnim, "The impact of data preprocessing on the performance of classification algorithms," *Procedia Computer Science*, vol. 194, pp. 35–42, 2021.

- [8] L. Zhang, S. Wang, and X. Liu, "Understanding bag-of-words model: a statistical framework," in *International Conference on Image and Graphics*, pp. 638–647, Springer, 2010.
- [9] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [10] F. Al-Hawari, A. Al-Badarneh, S. Khasawneh, and H. Barham, "A machine learning-based help desk system for IT service management," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 701–712, 2021.
- [11] S. Ghosh and S. Choudhury, "Impact of data preprocessing on the performance of text classification: An empirical study," *International Journal of Computer Applications*, vol. 56, no. 12, pp. 1–6, 2012.
- [12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Elsevier, 2011.
- [13] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- [14] A. Parameswaran, K. Patel, and R. Mishra, "Custom stopwords list generation for domain-specific text classification," *Procedia Computer Science*, vol. 185, pp. 471–478, 2021.
- [15] M. M. Rahman, M. R. Islam, and N. Akter, "Ticket classification using machine learning for IT service management," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 4, 2018.
- [16] A. Joshi, R. Sinha, and P. Sharma, "Automated IT ticket categorization using deep learning," *IEEE Access*, vol. 8, pp. 56045–56053, 2020.
- [17] N. Kumar and R. Singh, "Word cloud-based exploratory analysis and keyword filtering in NLP pipelines," *Procedia Computer Science*, vol. 152, pp. 135–142, 2019.
- [18] A. Garcí'a and S. Garcí'a, "Preprocessing in text mining: A survey," *Journal of Intelligent & Fuzzy Systems*, vol. 31, no. 1, pp. 1–11, 2016.
- [19] A. Kannan and V. Gurusamy, "A review on text preprocessing techniques for text mining applications," *International Journal of Computer Science & Engineering Technology*, vol. 5, no. 1, pp. 1–5, 2014.
- [20] S. Salloum, M. Al-Emran, and K. Shaalan, "Big data analytics on unstructured data: A review," *International Journal of Data Science and Analytics*, vol. 4, no. 2, pp. 85–104, 2017.
- [21] A. Kumar and J. Singh, "Automated classification of IT service tickets using machine learning," in *Procedia Computer Science*, vol. 152, pp. 218–225, Elsevier, 2019.
- [22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, 2019.
- [23] K. Saurabh, M. M. Tripathi, and S. Mahapatra, "Efficient utilization of energy in IoT devices using machine learning algorithms," *International Journal of Experimental Research and Review*, vol. 47, pp. 133–145, 2025.
- [24] J. Ahmad and S. Kumar, "Emotion classification from COVID-19 pandemic tweets using Roberta," *Journal of Information Systems Engineering and Management*, vol. 10, no. 32s, pp. 546, 554, 2025.
- [25] J. Ahmad and S. Kumar, "Sentiment analysis of COVID-19 tweets using BERT," *Educational Administration: Theory and Practice*, vol. 30, no. 5, pp. 5052–5059, 2024.
- [26] K. Saurabh, M. M. Tripathi, and S. Mahapatra, "A robust machine learning approach using smote and hyperparameter optimization for diabetes prediction using

- wearable IoT device data," *Journal of Dynamics and Control*, vol. 9, no. 2, pp. 44–56, 2025.
- [27] M. M. Tripathi and P. K. Patahk, "A systematic review: Forecasting post-pandemic health trends with machine learning methods," *International Journal of Intelligent Systems and Application in Engineering*, vol. 166, 2024.
- [28] A. Fawaz et al., "An automated help desk system for IT service management using machine learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 7, pp. 812–823, 2021.
- [29] GreyCampus, "Incident management."
<https://www.greycampus.com/opencampus/itil-foundation/incident-management>.
- [30] Sqlservertutorial, "SQL Server data types," 2020. <https://www.sqlservertutorial.net/sql-server-basics/sql-server-data-types>
- [31] W. Li and R. Zhang, "Language detection and filtering for multilingual IT support data," in *2020 IEEE International Conference on Big Data (Big Data)*, pp. 1234–1242, IEEE, 2020.
- [32] S. H. Kim and J. Lee, "Text normalization and preprocessing for machine learning applications in IT service datasets," *Expert Systems with Applications*, vol. 166, p. 114050, 2021.
- [33] B. Almeida and C. Silva, "Keyword extraction techniques for IT service tickets: A comparative study," *Information Processing & Management*, vol. 53, no. 6, pp. 1369–1381, 2017.
- [34] A. Singh and P. Sharma, "Data preprocessing techniques for improving text classification in IT service management," *Journal of Information Systems and Technology Management*, vol. 19, no. 1, pp. 1–15, 2022.
- [35] R. Studio and A. AWS, "Text mining using R," March 2020. <https://rstudio-pubs-static.s3.amazonaws.com>.
- [36] E. Guides, "Text mining and word cloud fundamentals in R: 5 simple steps you should know," November 2017. <https://www.r-bloggers.com/2017/02/text-mining-and-word-cloud-fundamentals>.
- [37] STHDA, "Text mining and word cloud fundamentals in R: 5 simple steps you should know," 2020. <http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals>.
- [38] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer, "Wrangler: Interactive visual specification of data transformation scripts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372, 2011.
- [39] L. Zhang, X. Wang, and R. Liu, "IT service ticket classification and routing using deep learning," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 509–521, 2020.
- [40] M. Alam and M. Hossain, "Automated IT ticket classification using machine learning techniques," *Journal of Network and Computer Applications*, vol. 181, pp. 103000, 2021.
- [41] Y. Liu and M. Chen, "Text preprocessing and feature engineering for ticket classification in IT service systems," in *2019 IEEE International Conference on Big Data*, pp. 4218–4225, IEEE, 2019.
- [42] J. Tang and P. Li, "Natural language processing techniques for IT service management: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–36, 2022.
- [43] W. Huang and L. Zhao, "Impact of preprocessing and feature selection on machine learning models for ticket categorization," *Expert Systems with Applications*, vol. 166, p. 114084, 2021.

- [44] H. Sun and Q. Li, "Enhanced ticket routing in IT service management using NLP and classification algorithms," *Information Sciences*, vol. 512, pp. 1292–1308, 2020.