

Automated Routing of Customer Complaints in Financial Services: NMF Topic Discovery Followed by Supervised Text Classification

Sohaib Ahmad (Corresponding Author)

Department of Data Science, York St John University, London Campus,
London, United Kingdom Email: sohaibahmed293@gmail.com

Umer Shahzad

University of the West of Scotland, London Campus, London, United Kingdom

Alina Akhtar

Department of Computer Science, University of Central Punjab, Lahore, Pakistan

ABSTRACT

Banks receive thousands of free-text complaints, and someone usually has to read each one before it can be sent to the right team. This study builds and tests a pipeline that does the sorting automatically. The main obstacle was the data: the complaint corpus had no usable labels, because its original product field spread the cases over 17 overlapping categories. The work therefore ran in two phases. In the first, 21,072 non-empty complaints were cleaned with NLTK, reduced to lemmatised nouns and converted into a TF-IDF matrix of 6,104 terms. Non-negative matrix factorisation (NMF) with five components then produced five readable topics: bank account services, credit card or prepaid card, theft or dispute reporting, mortgages and loans, and others. In the second phase, these topics served as labels for six classifiers trained on 75% of the data and tested on the remaining 5,268 complaints. Logistic Regression scored highest, with 95.54% accuracy and a weighted F1-score of 0.96, followed by LinearSVC (95.18%) and LightGBM (94.40%). Random Forest (88.40%), a single Decision Tree (84.55%) and Multinomial Naive Bayes (78.34%) trailed well behind; Naive Bayes found only 28% of the complaints in the "Others" class. The results show that a simple linear model on sparse TF-IDF features can route complaints accurately at very low cost. Because the labels come from NMF on the same features, the scores measure agreement with the topic model rather than with human judgement, and the paper discusses what this means for deployment.

Keywords: Natural Language Processing; Customer Complaints; Topic Modelling; Non-Negative Matrix Factorisation; Text Classification; TF-IDF; Financial Services

1. Introduction

Customers report to financial institutions via web forms, email and social media submissions, and nearly all of which are free text. Most organisations will have a support agent who reads the complaints and decides upon the issue and passes it on to a department, for example mortgages, cards or fraud. Slow and repetitive work and it is easy to make mistakes if the queue is big. A complaint received by a wrong team remains pending for a longer period of time and delayed responses are one of the most effective ways to lose the trust of a customer.

Natural language processing (NLP) is an easy way out. The first sorting step can take seconds if a model is able to read a complaint and predict the department it belongs to. The problem is that in order to be used, supervised models require labeled examples; and real complaint information is seldom labeled in the desired fashion. This is evident in the corpus used in this study. It does have a product field, but the cleaned data in that field has 17 categories, some of which have overlap, such as the Credit card and Credit card or prepaid card. Having it learn on such a target would teach it the irregularities of the original form, not the structure of the complaints.

This study thus views the task as two-intertwined problems. The first is unsupervised: Find a small number of coherent topics of complaints in unlabelled text and label them. The second is supervised: train and compare classifiers for the prediction of those topics for new complaints. The five areas of business to target were bank account, credit card or prepaid card, theft or dispute reporting, mortgages and loans, and a "Others" category. The following three research questions were used to guide the work:

RQ1. In this case, can part-of-speech filtering and lemmatisation of the raw text of a complaint using NLTK make a feature set of the right quality for machine-learning?

RQ2. What is NFM's accuracy for recovering 5 meaningful topics from the unlabelled corpus that are of interest for business?

RQ3. Which of the six supervised classifiers (Logistic Regression, LinearSVC,

LightGBM, Random Forest, Decision Tree and Multinomial Naive Bayes) is the most accurate at predicting these topics?

The paper has three contributions. It explains a full and repeatable workflow from scratch (raw complaint data) to a deployable classifier with no manual labelling. It evaluates six basic classifiers using the same split and gives results for each class, and not only for average accuracy. The applied work often neglects to be candid about what the scores mean, and here, it isn't. The related work is discussed in Section 2, the method in Section 3, the results in Section 4, and the discussion in Section 5 and the limitations and further work in Section 6.

2. Related Work

2.1 NLP in customer support

The first NLP systems used hand written grammars and symbolic rules. They were accurate but fragile and failed to generalize to the range of actual user language (Hirschberg & Manning, 2015). Many of these rules have been superseded by models that have been learnt from data as provided by statistical machine learning, and more recently by deep learning which enables systems to learn semantic and contextual representations directly from large corpora (Young et al., 2018). The economics were again upended by Transfer learning. Howard and Ruder (2018) demonstrated the fine-tuning of a pre-trained language model to a new classification task using a small amount of labelled data, and BERT's bidirectional transformer set a new bench mark on classification and intent detection (Devlin et al., 2019). These improvements have transformed systems from matching keywords to assistants that consider the context in customer service. This change in legal technology is similar to what Dale (2019) refers to as a change in legal practice, whereby the use of NLP helps in analysing and making decisions that previously could only be made by a specialist reader.

2.2 Topic modelling and unsupervised learning

Topic models are a natural step to take when a complaint archive lacks the reliable labels and are used to extract latent themes from unlabelled text. The most common techniques are grouped into two categories by Jelodar et al. (2019): probabilistic models, usually the benchmark for these methods is Latent Dirichlet Allocation (LDA) (Blei et al., 2003); and matrix factorisation methods, e.g. NMF (Lee & Seung, 1999). LDA is difficult to apply to

short, sparse documents, since it requires co-occurrence of words in each document in order to estimate the mixture of topics. Chen et al. (2019) compared the two families on short-text corpora and found that NMF generally produced more interpretable topics, which they attribute to its additive, parts-based decomposition. More recent work combines transformer embeddings with clustering. BERTopic, for instance, improves topic coherence by grouping documents in an embedding space before extracting topic words (Grootendorst, 2022). In either case, it is possible to use topic models as a pre-labelling procedure for supervised learning as is done in this study, NMF.

2.3 Supervised text classification

The majority of complaint classification systems are supervised. Kowsari et al. (2019) review all the available algorithms, including Naive Bayes, support vector machines, convolutional networks and recurrent networks, and highlight that TF-IDF features and linear models continue to be good baselines for many tasks. Linear SVMs have been well recognized to be a good choice for text because feature spaces are large and sparse, and most categories are near linearly separable (Joachims, 1998). The advantage of the multinomial Naive Bayes is that it is inexpensive and usually performs as well as the other models on short documents (McCallum & Nigam, 1998); tree ensembles like Random Forests (Breiman, 2001) and gradient-boosted trees like LightGBM (Ke et al., 2017) can be more costly, but they can capture non-linear interactions. In a recent review, Minaee et al. (2021) show that transformer models currently outperform most other deep learning classifiers on most benchmarks, but require significantly more computation and they are less interpretable than linear models.

2.4 Explainability, ethics and fairness

There are concerns about transparency and bias with the automation of customer interactions. According to Barredo Arrieta et al. (2020), explainability should be built into AI systems from the outset – not retrofitted on. Binns (2018) connects algorithmic fairness to political philosophy and to questions of accountability when they make errors in their decisions, while Mehrabi et al. (2021) list the various kinds of bias that data-driven systems can absorb and reproduce such as representation bias and historical bias. These concerns are of importance in complaints management as a case may be mis-routed, which can cause further delay in redress for a complainant who is already frustrated. Other regulations like those of the EU

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

General Data Protection Regulation also impose a need for the organisations to be able to account for decisions made with the help of models that can be considered, and to ensure that a human is involved when an automated decision is not certain. The main studies referred to in this section are summarised in Table 1.

Table 1. Summary of key studies informing this work

Study	Focus	Approach	Relevance to this study
Young et al. (2018)	Deep learning trends in NLP	Survey of CNN, RNN and sequence models	Traces the move from rule-based to learned representations
Howard & Ruder (2018)	Transfer learning for text classification (ULMFiT)	LSTM pre-training and fine-tuning	Shows pre-trained models adapt to small labelled sets
Devlin et al. (2019)	Contextual representations (BERT)	Transformer pre-training	Benchmark for future deep models on this task
Jelodar et al. (2019)	Topic modelling survey	Review of LDA and related models	Taxonomy and evaluation criteria for topic discovery
Chen et al. (2019)	LDA versus NMF on short text	Experiments on short-text corpora	Supports NMF for sparse complaint text
Kowsari et al. (2019)	Text classification algorithms	Survey of classical and deep models	Supports TF-IDF with linear classifiers as a baseline
Minaee et al. (2021)	Deep learning text classification	Comparative review	Places transformers as the current accuracy leaders
Grootendorst (2022)	BERTopic	Transformer embeddings with clustering	Alternative to NMF for future work
Barredo Arrieta et al. (2020)	Explainable AI	Conceptual taxonomy	Argues for interpretable models in deployment
Binns (2018); Mehrabi et al. (2021)	Fairness and bias in machine learning	Theoretical analysis; survey	Frames the ethical risks of automated routing

Source: compiled by the authors.

2.5 Research gap

There are 3 points of weakness. First, studies usually focus on topic modeling and classification separately, and only a few include a description of a pipeline from an unlabelled archive to a deployable classifier. Secondly, the accurate deep models are costly and elusive, reducing their applicability for the day-to-day routing in regulated organisations. Third, applied papers will only frequently give overall accuracy, which may mask a low accuracy on smaller but key categories like fraud and disputes. To this end, this study discusses the NMF-plus-classifier pipeline that facilitates the interpretability of the resulting NMF components, compares the performance of six classifiers on the same splitting, and provides per-class analysis.

3. Methodology

Figure 1 gives an overview. In phase 1, raw complaints are labeled using preprocessing, TF-IDF weighting and NMF. In Phase 1, he trains classifiers with those labels on the same training set and tests them on a separate test set in Phase 2. All steps were implemented in Python with pandas, NLTK (Bird et al., 2009), scikit-learn (Pedregosa et al., 2011) and LightGBM, and every random step used a fixed seed (random_state = 42).

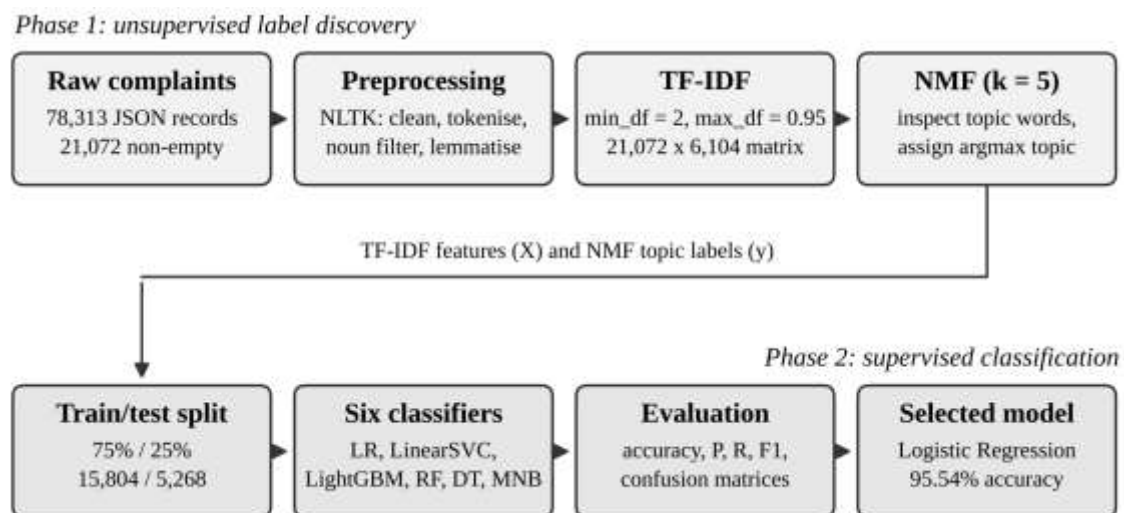


Figure 1. Two-phase pipeline for complaint topic discovery and classification Source: developed by the authors.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

3.1 Dataset and preparation

The data consists of a publicly available set of 78,313 customer complaints on a large retail bank, distributed as a JSON export. A set of data in each record includes multiple fields. These are two that were used: `complaint_what_happened` (renamed `complaint_text`) is the customer's own account of the problem, and `product` is the category recorded at the time of the complaint. The JSON was converted to a pandas data frame. The JSON data was converted into a pandas data frame. Many complaints lacked a narrative and deleting these records left 21,072 complaints for analysis.

The reason why the `product` field was not a target is represented in Figure 2. Its 17 values are badly imbalanced, ranging from 5,039 complaints for "Credit card or prepaid card" down to a handful for "Payday loan", and several values describe the same product under different names. The field was used only for exploration.

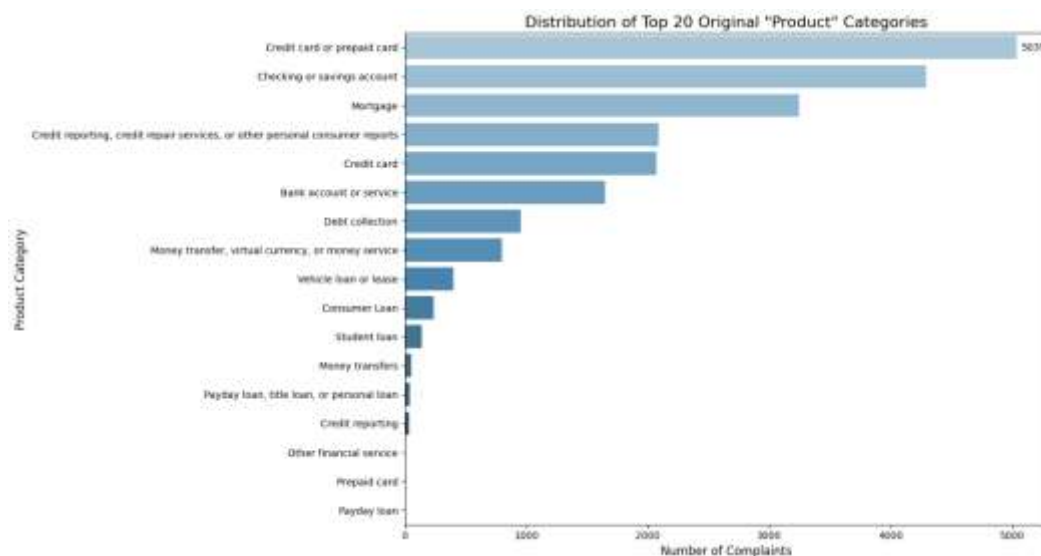


Figure 2. Distribution of the original product categories in the cleaned corpus (N = 21,072) Source: authors' analysis.

3.2 Text preprocessing

Each complaint passed through the same sequence of steps. The text was converted to lower case. Anonymisation placeholders such as "xxxx" were removed with a regular expression, and punctuation, digits and other non-alphabetic characters were stripped. The cleaned text

was tokenised with NLTK and English stop words were removed. Each remaining token was then tagged for part of speech, and only nouns and proper nouns of more than one character were kept. These were lemmatised with NLTK's WordNet lemmatiser (Miller, 1995), so that "cards" became "card" and "payments" became "payment". Finally the lemmas were joined back into a single string per complaint. Two design choices deserve explanation. Lemmatisation was preferred to stemming. A stemmer cuts word endings by rule and often produces fragments that are not words, such as "investig" for both "investigation" and "investigating", whereas a lemmatiser uses a dictionary to return the base form (Manning et al., 2008). Cleaner word forms give NMF more reliable co-occurrence patterns to work with. The noun filter was the second choice. Complaints are full of emotional adjectives and adverbs, such as "terrible", "angry" and "never", which say a lot about the customer's mood but little about which department should respond. Keeping only nouns concentrates the vocabulary on the things the complaint is about: transactions, fees, cards, accounts and loans.

3.3 Exploratory analysis

Word frequencies in the processed text were checked before modelling. Among single words, "chase", "account", "credit", "card" and "bank" were the most frequent, and "credit card" was by far the most common two-word phrase, with 12,849 occurrences (Figure 3). The top three word combination was chase credit card (2,153 times). The 100 most common words were summarized in a word cloud, which also conveyed the same message. These checks revealed no significant differences in the substantive vocabulary, and indicated that cards and accounts, and payments and loans would each be different themes for the target language.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

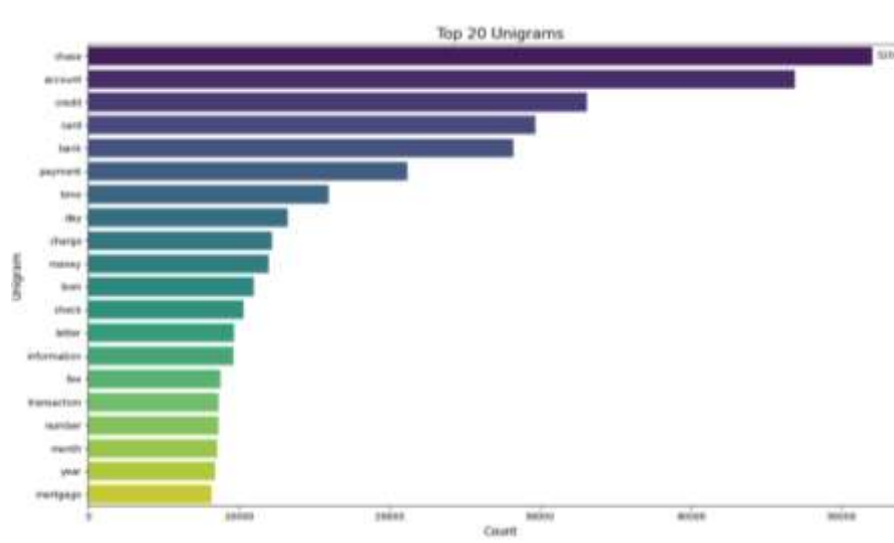


Figure 3(a). Top 20 unigrams (top) after preprocessing Source: authors' analysis.

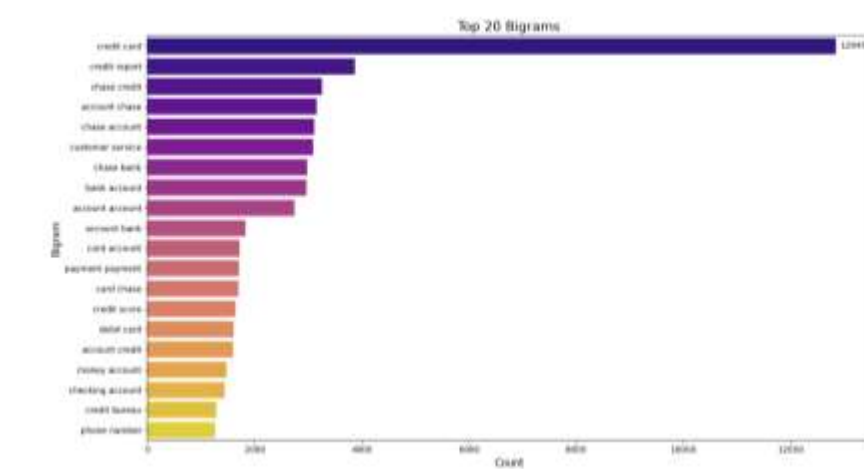


Figure 3(b). Top 20 bigrams (bottom) after preprocessing Source: authors' analysis.

3.4 Feature extraction with TF-IDF

The processed text was transformed to a term frequency-inverse document frequency (TF-IDF) matrix that is required by machine learning models. The term frequency $tf(t,d)$ is the number of times that the term t occurs in document d divided by the length of d :

$$TF(t, d) = f_{t,d} / \sum_k f_{k,d} \quad (1)$$

Inverse document frequency assigns more weight to terms with few occurrences in the

documents. With N documents, of which $DF(t)$ contain the term,

$$IDF(t) = \log(N / DF(t)) \quad (2)$$

and the weight of term t in document d is the product of the two:

$$TF-IDF(t, d) = TF(t, d) \times IDF(t) \quad (3)$$

A term therefore scores highly when it is common in one complaint but rare across the corpus, which is what makes it useful for telling complaints apart (Manning et al., 2008). Scikit-learn's `TfidfVectorizer` was used with `min_df = 2`, which drops terms found in only one document (mostly typos), `max_df = 0.95`, which drops terms found in more than 95% of documents, and its built-in English stop-word list as a second filter. Scikit-learn's implementation also applies smoothing and L2 normalisation by default. The result was a sparse document-term matrix of 21,072 complaints by 6,104 terms.

3.5 Topic discovery with NMF

NMF approximates the non-negative document-term matrix X (n documents $\times$ m terms) as the product of two smaller non-negative matrices (Lee & Seung, 1999):

$$X \approx WH, \quad W \in \mathbb{R}_+^{n \times k}, \quad H \in \mathbb{R}_+^{k \times m} \quad (4)$$

where W gives the weight of each of the k topics in each complaint and H gives the weight of each term in each topic. The factors are found by minimising the reconstruction error $\|X - WH\|_F^2$ subject to non-negativity. Because nothing can be subtracted, each topic is built by adding words together, which tends to give topics that people can read and name. NMF was fitted with $k = 5$ to match the five target categories. The 15 highest-weighted words in each row of H were inspected, a business name was assigned to each topic, and every complaint was labelled with the topic that had the largest value in its row of W .

3.6 Classifiers and training

The labelled data was split at random into a training set of 15,804 complaints (75%) and a test set of 5,268 complaints (25%). The TF-IDF matrix served as the feature set and the NMF topic as the target. Six classifiers were trained on the same training data. Three were specified in the original project proposal (Logistic Regression, Decision Tree and Random Forest) and three were added for comparison (Multinomial Naive Bayes, LinearSVC and LightGBM). To

keep the comparison fair, all were run with library defaults and no hyperparameter tuning; Table 2 lists the settings that were set explicitly.

Table 2. Classifiers compared and their configuration

Classifier	Family	Configuration and rationale
Logistic Regression	Linear, probabilistic	solver = liblinear; random_state = 42. Fast, interpretable baseline for sparse, high-dimensional text.
LinearSVC	Linear, margin-based	random_state = 42; default regularisation. Finds the maximum-margin separating hyperplane; well suited to sparse TF-IDF data (Joachims, 1998).
LightGBM	Gradient-boosted trees	random_state = 42; n_jobs = -1. Grows trees leaf-wise, each correcting the errors of the last (Ke et al., 2017).
Random Forest	Bagged tree ensemble	random_state = 42; default 100 trees. Averages many decorrelated trees to reduce overfitting (Breiman, 2001).
Decision Tree	Single tree	random_state = 42; unpruned. Highly interpretable but prone to overfitting.
Multinomial Naive Bayes	Probabilistic	Default smoothing (alpha = 1.0). Standard text baseline that assumes conditionally independent features (McCallum & Nigam, 1998).

Source: authors' implementation.

3.7 Evaluation

Models were evaluated on the test set only. Accuracy, the share of complaints assigned to the correct topic, was the headline measure. Because the topics are imbalanced, precision, recall and F1-score were also computed for each class. With TP, FP and FN denoting true positives, false positives and false negatives for a given class,

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}), \quad \text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (5)$$

$$\text{F1} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall}) \quad (6)$$

Macro averages weight each class equally, while weighted averages weight each class by its number of test complaints. Confusion matrices were drawn for the strongest models to show which topics were mistaken for which.

4. Results

4.1 Discovered topics

NMF converged and produced five topics with clearly different vocabularies (Table 3). Topic 4 is plainly about property lending, with "loan", "mortgage", "home", "modification" and "foreclosure" among its top words. The first topic is about credit cards, and has words like "card", "credit", "limit", "score", and "inquiry". Topics 3 includes "charge", "transaction", "dispute", "claim" and "fraud", suggesting fraudulent and/or disputed transactions, while Topic 0 refers to normal account transactions like checks, deposits, transfer and branch visits. Topic 2 has the least focus. It has a combination of several smaller products such as the words "payment", "amount", "fee", "statement", "auto", "debt", hence the name "Others". The remainder "xxxx" in this topic is a legacy of the anonymised dates which passed the placeholder filter, which is discussed in Section 6.1.

Table 3. Highest-weighted words for each NMF topic and the label assigned

Topic 0: Bank account services	Topic 1: Credit card / prepaid card	Topic 2: Others	Topic 3: Theft / dispute reporting	Topic 4: Mortgages / loans
account	card	chase	chase	loan
bank	credit	xxxx	charge	mortgage
check	chase	payment	transaction	home
money	account	amount	dispute	payment
fund	balance	fee	claim	modification
deposit	limit	year	fraud	year
day	score	month	email	time
branch	inquiry	day	information	chase
transaction	report	money	account	letter
number	company	statement	purchase	document

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Topic 0: Bank account services	Topic 1: Credit card / prepaid card	Topic 2: Others	Topic 3: Theft / dispute reporting	Topic 4: Mortgages / loans
information	application	service	letter	property
service	letter	check	call	information
call	information	time	item	foreclosure
reason	point	auto	number	bank
transfer	business	debt		

Note: The Top 15 most frequent words per topic are presented (the remaining ones were not included); repeated occurrences in the Topics 3 and 4 Top Wordlists were not included. Source: authors' analysis.

The topics are unevenly sized (Figure 4). Credit card and bank account complaints are the two largest groups, with roughly 6,000 complaints each, followed by mortgages and loans, theft and disputes, and finally "Others". The test set mirrors this: 1,517 credit card complaints (28.8%), 1,489 bank account complaints (28.3%), 941 mortgage and loan complaints (17.9%), 704 theft and dispute complaints (13.4%) and 617 "Others" (11.7%). Any classifier has to cope with categories that differ in size by a factor of about 2.5.

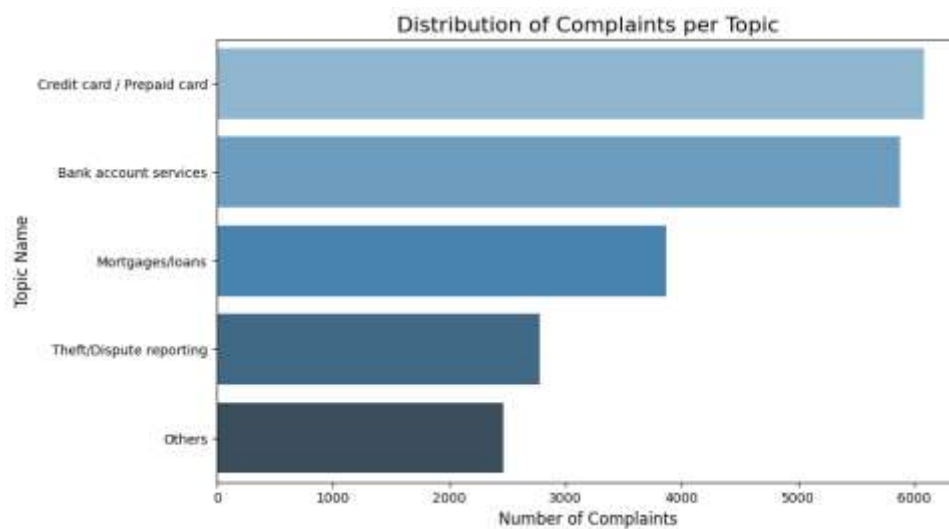


Figure 4. Number of complaints assigned to each NMF topic Source: authors' analysis.

4.2 Overall classifier performance

Table 4 and Figure 5 compare the six classifiers. They fall into three groups. The two linear

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

models lead, with Logistic Regression at 95.54% and LinearSVC at 95.18%. LightGBM comes close behind at 94.40%. The remaining three models are clearly weaker: Random Forest reached 88.40%, the single Decision Tree 84.55% and Multinomial Naive Bayes 78.34%. On the 5,268 test complaints, the gap between the best and worst model amounts to about 910 extra misrouted cases.

Table 4. Overall performance on the test set (n = 5,268)

Classifier	Family	Accuracy (%)	Macro F1	Weighted F1
Logistic Regression	Linear	95.54	0.95	0.96
LinearSVC	Linear	95.18	0.95	0.95
LightGBM	Boosted trees	94.40	n.r.	n.r.
Random Forest	Bagged trees	88.40	0.87 ^a	0.88 ^a
Decision Tree	Single tree	84.55	n.r.	n.r.
Multinomial Naive Bayes	Probabilistic	78.34	0.73	0.77

Note: n.r. = not reported in the original analysis. ^a Computed from the Random Forest confusion matrix in Figure 6. Source: authors' analysis.

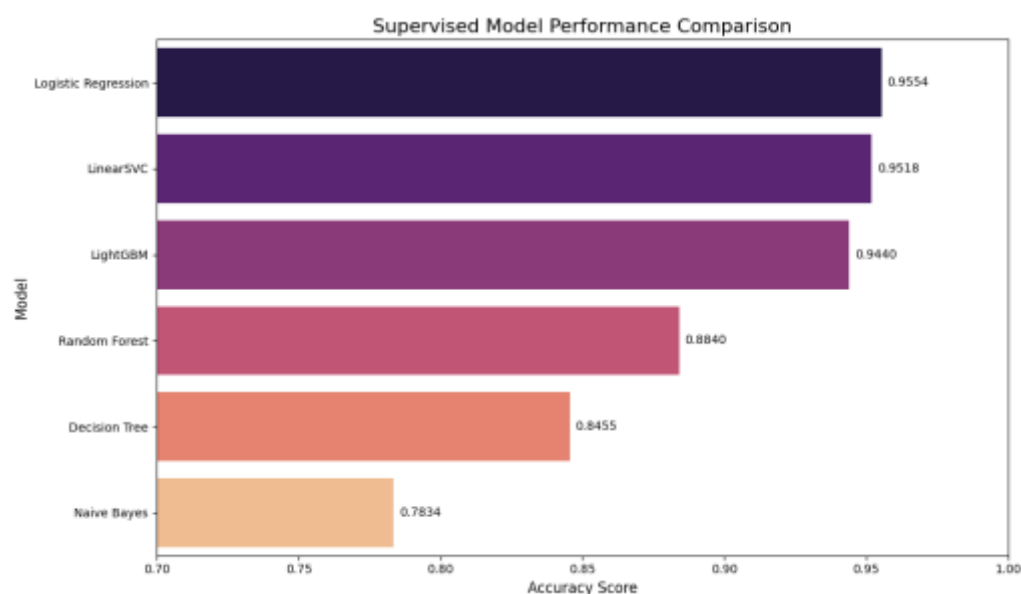


Figure 5. Test accuracy of the six classifiers

Note: The horizontal axis starts at 0.70. Source: authors' analysis.

4.3 Per-class performance

Overall accuracy can hide weak classes, so Table 5 breaks the results down by topic. Logistic Regression performed well in all of the regions, with an F1-score of 0.94 in regions of theft and dispute reporting, and an F1-score of 0.96 in regions of bank accounts and credit cards. The lowest recall was for "Others" and theft and disputes, and in both these cases the precision of its predictions was 0.97 or greater, meaning that it was usually correct when it did make a prediction. The other model, LinearSVC, was very alike, with F1-scores of 0.94 to 0.96. Naive Bayes was a different story. It classified a significant number of complaints as credit card or bank account concerns, resulting in above average recall percentages on these two categories (0.90 and 0.88), but it paid a high price for doing so on other categories. It was able to resolve only 28% of the "Others" complaints (F1 = 0.42) and 64% of the theft and dispute complaints. Random Forest was somewhere in the middle, having high precision and low recall for the same two minority classes.

Table 5. Per-class precision (P), recall (R) and F1-score on the test set

Topic	LR P	LR R	LR F1	SVC F1	RF R	RF F1	NB R	NB F1	Support (n)
Bank account services	0.95	0.97	0.96	0.95	0.94	0.91	0.88	0.83	1,489
Credit card / prepaid card	0.94	0.98	0.96	0.96	0.94	0.89	0.90	0.80	1,517
Others	0.98	0.91	0.95	0.95	0.69	0.80	0.28	0.42	617
Theft / dispute reporting	0.97	0.91	0.94	0.94	0.81	0.85	0.64	0.74	704
Mortgages / loans	0.96	0.95	0.95	0.95	0.90	0.91	0.87	0.85	941

Note: LR = Logistic Regression, SVC = LinearSVC, RF = Random Forest, NB = Multinomial Naive Bayes. All values are calculated from the classification reports except for those in Random Forest where they are computed from its confusion matrix. Source: authors' analysis.

4.4 Confusion matrices

Figure 6 shows the confusion matrices for the two linear models and Random Forest. For Logistic Regression, recall on the diagonal ranges from 91.2% ("Others") to 98.0% (credit cards), and no single off-diagonal cell exceeds 4.5%. The largest error is 32 theft and dispute

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

complaints predicted as credit card complaints, which makes sense, since many disputes concern card transactions. LinearSVC is nearly the same, with a diagonal recall of between 92.5% and 97.1%. Random Forest commits the same type of errors, but in greater numbers. It refers 13.1% of theft and dispute cases (92) to the credit card class and only recovers 69.2% of the casualties in the "Others" class, with the remainder split across bank accounts (11.5%), credit cards (9.1%) and mortgages (7.5%). The expensive mistakes in practice are a report of a fraud to the card team or a random complaint tossed from one team to the next.

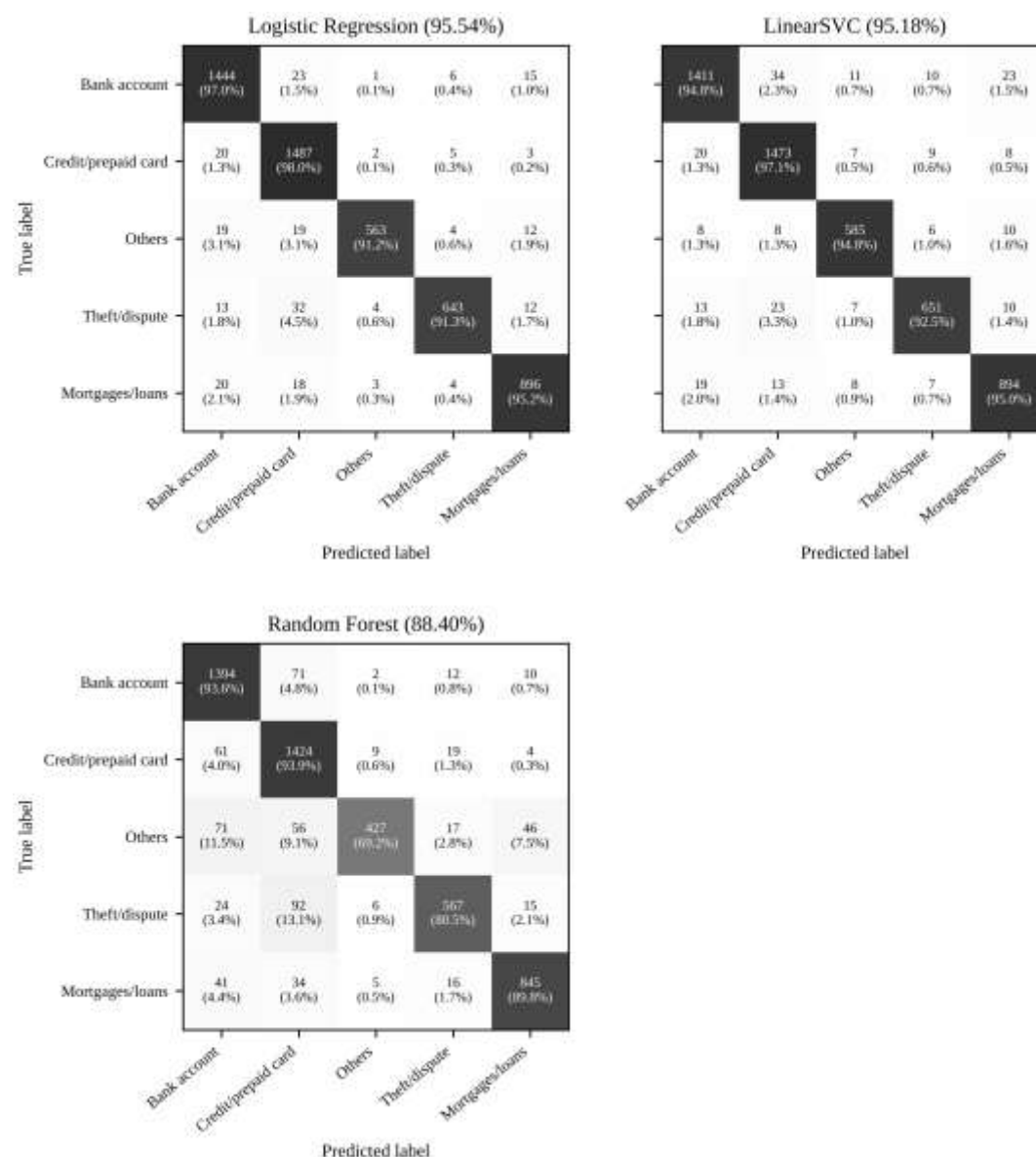


Figure 6. Confusion matrices for Logistic Regression, LinearSVC and Random Forest

Note: Row percentages are shown (recall on the diagonal). Source: authors' analysis.

5. Discussion

5.1 What the topics tell us

NMF carried out its duties on RQ2, as requested. The five topics are aligned with actual business functions, rather than abstract word clumps, and four of the topics could be named with great certainty from the top words. This is consistent with Chen et al. (2019), who determined that NMF was appropriate for short and sparse text like this. The weak point is the "Others" topic. It consists of payments, fees, statements, auto loans and debt and a bank would likely want to break it down even more before sending it out for processing. The size of the topics is also important. A model that was just good at the two largest classes could be reasonably good but not do well on the smaller classes, which made per-class results a necessity.

5.2 Why the linear models won

On RQ3, Logistic Regression and LinearSVC came out on top, and the margin between them (0.36 percentage points) is too small to matter in practice. It isn't surprising they're successful. Here the features space is quite large and sparse – in this case the space is larger than 6,000 – and often the classes are close to linearly separable (Joachims, 1998; Kowsari et al., 2019). There is also a more specific reason as is discussed in Section 6.1. The labels were produced by NMF from the same TF-IDF matrix, and each complaint was assigned to the topic with the largest weight in an essentially linear decomposition of its feature vector. A linear classifier is a good candidate to learn the above mapping. Logistic Regression is a more useful technique for a bank than accuracy. It trains in seconds, it outputs class probabilities that can be used for flagging uncertain cases, and the coefficients it produces indicate which words influence a complaint to move it towards each department, which is transparent enough to please regulators (Barredo Arrieta et al., 2020).

5.3 Boosting versus bagging

In fact, lightGBM achieved 94.40% accuracy without any tuning, just one percent point

behind the linear models and six points above than Random Forest. Boosting grows then grows, and iterates over the trees to add trees which fit the errors of the previous ensemble, while Random Forest grows its trees all independently and averages them. The few high-frequency words that are informative for a minority topic may not be included in the randomly selected feature subsets for Random Forest, which could be the reason for its low recall on the minority topics. If the data in the future is found to require non-linear decision boundaries, using LightGBM is still a reasonable choice, as it does not give any accuracy advantage when working with the current data and is less easy to explain.

5.4 Why Naive Bayes struggled

Multinomial Naive Bayes is typically a good text baseline model (McCallum & Nigam, 1998), and in this case has an accuracy of 78.34% - so why is this? This corpus is a poor fit for the model, which assumes that words are independent given the class. Words like chase, account, card, and payment are repeated in almost every topic and have many supporting evidences which help bring the predictions towards the largest classes. The classification report confirms this: high recall for credit card's class and bank account's classes, and almost nonexistent for 'Others'. With TF-IDF weights rather than raw counts as input, the multinomial likelihood is also only an approximation, which may add to the problem. The single Decision Tree achieved the typical pitfall of fitting noise in the training data of an unpruned tree (84.55%).

5.5 The role of preprocessing

The frequency analysis resulted in a smaller vocabulary of 6,104 terms for RQ1, the processing chain generated, where the predominant concepts of the business concepts were the main ones. Filters out a lot of the emotional words from complaints and leaves only nouns in the models, which are the things that make the decision on how to route: cards, accounts, fees, transactions, loans. The contribution of the noun filter to accuracy in this study is not known because there was not a pipeline that retained all words. However, it is likely that the narrower vocabulary was conducive to the formation of clean topics, and the high classification scores were possible because of the clean topics.

5.6 Implications for practice

The primary lesson of a support operation is that a lot of expensive infrastructure is not required for proper routing of the call at the first call. Training a Logistic Regression model on TF-IDF features can be done on a laptop, it is served through a simple API and can be retrained in a matter of hours. Two protections would still have to be in place prior to deployment. First, predictions that have low probabilities should not go directly to a department, but rather should first be directed to a human agent, thereby ensuring that a person is there for the cases that the prediction is difficult to make (Binns, 2018). Secondly, the topic definitions need to be checked with the teams receiving the complaints—the labels created by an algorithm are only useful if they align with the teams within the organisation's structure.

6. Conclusion

This study has developed a two-stage pipeline which transforms an unlabelled archive of bank complaints into an automatic routing model. NMF on a noun-only TF-IDF representation identified five meaningful topics: Bank account services, Credit cards, Theft and dispute, Mortgage and loan, and a residual topic. Then six classifiers were trained to predict these topics. Logistic Regression achieved the highest accuracy of 95.54% and the highest weighted F1-score of 0.96 of all models, and performed well on the smaller topics, closely followed by LinearSVC at 95.18%. LightGBM outperformed the other classifiers, with a score of 94.40%, with the other classifiers (Random Forest, Decision Tree and Multinomial Naive Bayes) lagging behind primarily due to their inability to classify many complaints in the minority classes. Considered together, the results indicate that a well-preprocessed TF-IDF-based representation, an NMF labelling phase and a default linear classifier provide a solid starting point for the automated triage of complaints that is fast, transparent and accurate.

6.1 Limitations

Several caveats qualify these results: The most crucial is the labelling. These were created by NMF on the same features as the classifiers later used, and NMF was adapted to a complete corpus before the train-test split. The reported accuracy hence is not a measure of how well each classifier matches the judgement of the human agents, but how well the topic model is reproduced; some information about the test documents was available during the time the labels were produced. To measure real routing accuracy a sample of complaints marked by employees would be required. Second, they are all based on a single set of data from a single

institution; topics and scores may not be generalizable to other banks or sectors without retraining. Third, because of the business requirement, a maximum of five topics were set; coherence measures could be applied to select a more principled value of k . Fourth, models were compared using one split with default settings, without any cross-validation or tuning, therefore small differences between the top models should not be interpreted. Fifth, some anonymisation placeholders remained after pre-processing in topic of "Others" (the token "xxxx") and the threshold of the filter needs to be narrowed. Lastly, TF-IDF is not context aware, making it possible for complaints with uncommon terms for a familiar issue to be misrouted.

6.2 Future work

There are three natural extensions. The first is to train a pre-trained transformer model (e.g. BERT or RoBERTa) on the labels provided by humans and evaluate the performance against the Logistic Regression baseline model while considering costs and explainability of any performance gains (Devlin et al., 2019). The second is deployment: if the model were wrapped as a REST API that is attached to a real helpdesk, routing suggestions would show up as soon as any complaint would hit the desk and it would be possible to measure the impact on resolution time. The third is a loop of activity learning (Settles, 2009). Since the complaints that were classified by low confidence would be sent for manual review, the revised labels would then be incorporated into the labelling set and the system would be able to classify new types of complaints without a complete manual relabelling.

DECLARATIONS

Acknowledgement: This article is derived from the first author's MSc Data Science dissertation at York St John University, London Campus. The authors thank Dr Sahar Ahmadzadeh for supervision and feedback throughout the project.

Funding: No funding body in the public, private, or nonprofit sectors provided a particular grant for this research.

Availability of data and material: The complaint corpus is publicly available. The code used for preprocessing, topic modelling and classification is available from the corresponding author on reasonable request.

Authors' contributions: Sohaib Ahmad designed the study, implemented the pipeline, ran the experiments and wrote the original draft. Umer Shahzad contributed to the analysis and interpretation of results and reviewed and edited the manuscript. Both authors approved the final version.

Conflicts of Interest: The author declare no conflict of interest.

Consent to Participate: Yes

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

Consent for publication and Ethical approval: The study used a public, anonymised dataset and involved no human participants.

REFERENCES

- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115. <https://doi.org/10.1016/j.inffus.2019.12.012>
- Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, PMLR 81, 149–159.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Chen, Y., Zhang, H., Liu, R., Ye, Z., & Lin, J. (2019). Experimental explorations on short text topic mining between LDA and NMF based schemes. *Knowledge-Based Systems*, 163, 1–13. <https://doi.org/10.1016/j.knosys.2018.08.011>
- Dale, R. (2019). Law and word order: NLP in legal tech. *Natural Language Engineering*, 25(1), 211–217. <https://doi.org/10.1017/S1351324918000475>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure (arXiv:2203.05794). *arXiv*. <https://doi.org/10.48550/arXiv.2203.05794>
- Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245), 261–266. <https://doi.org/10.1126/science.aaa8685>
- Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 328–339. <https://doi.org/10.18653/v1/P18-1031>
- Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey. *Multimedia Tools and Applications*, 78(11), 15169–15211. <https://doi.org/10.1007/s11042-018-6894-4>
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98 (Lecture Notes in Computer Science, Vol. 1398, pp. 137–142)*. Springer. <https://doi.org/10.1007/BFb0026683>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, 3146–3154.

Online ISSN

3007-3197

Print ISSN

3007-3189

<http://amresearchreview.com/index.php/Journal/about>

- Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4), Article 150. <https://doi.org/10.3390/info10040150>
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791. <https://doi.org/10.1038/44565>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization* (pp. 41–48). AAAI Press.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6), Article 115. <https://doi.org/10.1145/3457607>
- Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11), 39–41. <https://doi.org/10.1145/219717.219748>
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, 54(3), Article 62. <https://doi.org/10.1145/3439726>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Settles, B. (2009). *Active learning literature survey* (Computer Sciences Technical Report 1648). University of Wisconsin-Madison.
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), 55–75. <https://doi.org/10.1109/MCI.2018.2840738>

Appendix A. Core Preprocessing Function

The function below was applied to every complaint to produce the processed text used for TF-IDF and NMF.

```
def normalize_text_nltk(text):
    text = str(text).lower()
    text = re.sub(r"\b[xX]+\b", "", text)          # anonymisation placeholders
    text = re.sub(r"^[a-z\s]", "", text)          # punctuation and digits
    text = re.sub(r"w*\d*w*", "", text)
    tokens = word_tokenize(text)
    lemmas = []
    for word, tag in nltk.pos_tag(tokens):
        if tag.startswith("N") and word not in stop_words and len(word) >
1:
            lemmas.append(lemmatizer.lemmatize(word, pos=get_wordnet_pos(t
ag)))
    return " ".join(lemmas)
```