A Benchmarking Tool for AI-for-Selection of Documents for Permanent Preservation

Santhilata Kuppili Venkata
The National Archives, UK

July 27, 2020

Abstract

It is important to categorise documents into retention schedules for the selection process. Usually, the documents are categorised using rules set by the various criterion of retention. These are handled by the knowledge and information management manually. However, the document selection and classification can be a daunting task with an ever-increasing volume of born digital documents. Also, the situation gets complicated with the ease with which the content of digital documents can be duplicated and reproduced. Multiple copies of the documents spread across various subdivisions within a department can pose a serious security threat if the document is not correctly categorised. The research team of digital preservation department at The National Archives (TNA), UK is researching on how to use existing artificial intelligence tools for the document selection process. This document reports the benchmarking tool developed by the team using open source libraries to compare with the popular commercial tools available in the market.

1 Problem Statement

Ever since the digital transformation took over the processes and services, the governments are using digital tools to improve their day-to-day activities and interactions with public [1, 2]. While the digitisation has helped the government departments to become agile organisations, the speed with which the volume of documents produced is causing a concern. Government departments are looking for newer technologies to handle their processes with digital document production. They are seeking the potential of Artificial intelligence (AI) and machine learning to help with large data volumes. The use of AI has become a new normal in every government system to face the high-volume digitization activities.

The selection of digital documents for permanent retention has become one such tedious issue faced by the knowledge and information management (KIM) team in the government departments. They need to categorize documents according to multiple criteria set to retain documents. With the growing volume of documents and their duplicates, KIM managers need the process to be automated. They need a complete pipeline to assign the retention categories based on a set of rules. Which means, the document selection for permanent retention is based on the assigned category for a document. Figure 1 refers to the activities that KIM would like to have in the workflow of the document selection tool. From the KIM's perspective, there are four important activities for document categorization as a closed loop feedback process described below.

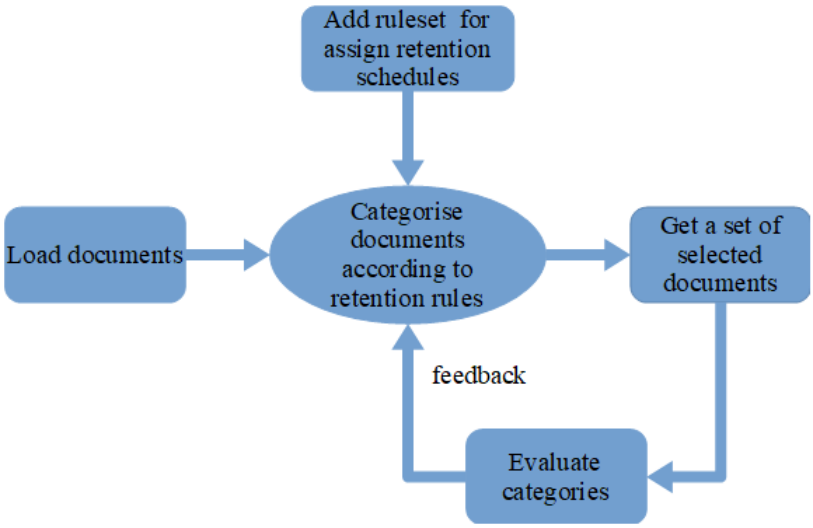

Figure 1: Activity diagram of the work flow (pipeline) to assign retention schedules for selection

- The **Load Documents** activity allows KIM team to add document corpus to be categorised. This activity defines number of ways that the documents can be uploaded to the tool. Example, it describes a plugin facility to upload a set of documents from a hard drive or downloaded from a cloud platform etc..
- The **Add rules** activity allows KIM to add rules set for retention schedules.
- The **Get the selected documents** activity is to collect suggestion of retention schedules for each of the documents.
- The **Evaluate retention categories** activity allows KIM team to inspect the categorisation and adjust or fine tune the and process by providing feed back.
- The **Categorisation of documents** process refers to the selection tool that loads the data into the system and trains the tool to assign retention categories to documents.

The **AI for Selection** project has been initiated to recommend existing tools for the selection process to government departments. We have created a benchmarking tool using open source libraries to evaluate the tools recommended. This document describes the development of the benchmarking tool in the following sections. The section 2 details the characteristics of the data. Some insights into the raw data and preprocessing to enable the data to fit into a model is described in section 5.1. Data modeling is described in section 5. Section 5.3 narrates the evaluation of the best model suitable for the document classification process.

2 Description of the document data

It is a common practice for the government organisations to have an Electronic Document and Records Management System (EDRMS) where, documents are arranged in a structured way. Also, organisations use shared drives (common document pools) as a temporary storage for sharing of the

documents that are used for day-to-day activities. Usually the common folders are unorganised. Organisations need to classify documents and select for permanent preservation in the shared drives with the same set of rules applied to EDRMS data. To reflect this scenario, we received two sets of documents from the repositories owned by TNA for the classification of documents into various retention categories. Also, we received 'the rules of retention' developed by the knowledge and Information team for the classification process. We received the raw document data corpus in a password protected hard drive.

The first set of documents are labelled with appropriate retention schedules based on the ruleset described by KIM. We have received an Excel sheet with detailed description of each document (metadata) along with their retention category. We will refer them as labelled documents here after. The second set of documents are sampled from random collections of a shared document pool (X drive of TNA). They are not labelled with retention schedules. We will refer them as unlabelled data from now on. However, the unlabelled documents are broadly categorised by the digital archiving department. In all, there are 118,677 documents of labelled documents from various departments of TNA, spread across **twenty** retention schedule categories in the first set. There are 50,000+ unlabelled documents in the second set. We set the second set for testing. Also, we have received the list of train and validation division of the labelled documents in 80:20 ratio. This is to compare the results of benchmarking tool with that of other suppliers. It is important to mention that

- *a document is classified with one retention schedule only.*
- *only few retention schedules are selected for permanent preservation.*

The document retention schedule categories are: 02, 03, 04, 05, 06, 07, 10, 11, 15, 15b, 16, 20, 21, 23, 24, 24a, 24b, 25, 27, 28, 32, 33. Out of the above categories, documents belonging to 04, 06, 15b, 17, 21, 33 are only selected for permanent retention and the remaining categories are to be deleted after varied number of years from the date of creation of the document. Hence documents are further classified into selected / Not selected categories for permanent preservation.

2.1 Loading of data

Since the benchmarking tool is to be developed within the secured peripheries of TNA, we did not consider developing various plugins to upload documents to the pipeline. The data is accessed from the hard disk for the rest of tasks in the pipeline.

2.2 Assigning ruleset for classification

The rules used for document classification are a set of complex conditions described by KIM. We chose to use machine learning methodology to develop the benchmarking tool instead of rule based engine. Hence the benchmarking tool made use of retention schedules provided in the labelled data for classification.

2.3 Document selection

We followed a two step process for the document selection. First, classify documents according to the retention schedule categories using machine learning models. Then further divide them into two

classes:

1. **Selected** for permanent preservation
2. **Not selected** for permanent preservation

3 Data preparation and insights into data

Technically different pre-processing techniques are to be applied to text based and media documents. The retention schedules are largely applicable to text based documents only. So in order to apply rules uniformly, we restricted the document classification for text data only. We selected 92,000+ documents with extensions .doc, .docx, .rtf, .pdf, .txt, .msg, .mbox, .xls, .xlsx from the corpus.

All important metadata about the files were extracted and compared with the metadata provided for the model training. The metadata fields selected for model training are:

- the file path,
- repository from where the document was sampled,
- author of the file,
- file size,
- retention schedule for the preservation policy,
- time last modified and
- the top most folder of the file structure.

On further examination, we have omitted the author and time last modified fields as they were over written and got corrupted while transferring data from its source to the experimentation site.

3.1 Exploratory data analysis

Some insights into the data are as follows:

The Figure 2 explores the number of documents present in the corpus within each repository. The figure shows the division of selected (orange) to not selected (blue) documents. There are 32 repositories in total. A correlation can be observed between the repository to selection of files for permanent preservation. For example, we can see that all documents from the digital preservation repository are selected for permanent preservation. Similarly, a large number of documents from Government audience are selected for permanent preservation category.

The number of documents present in each of the retention schedule category are visualised in the Figure 3. It represents the division of selected (orange) to not selected (blue) documents in each of the retention category. There are only four categories (04, 06, 21 and 33) of permanent preservation are present in the labelled data. Also, it appears that the distribution of documents with respect to retention categories is highly imbalanced. Since imbalanced distribution of classes have huge impact on the classification tasks, we had to balance classes by the methods of data augmentation [3]. Balancing classes protect minority classes from mis-classification errors.

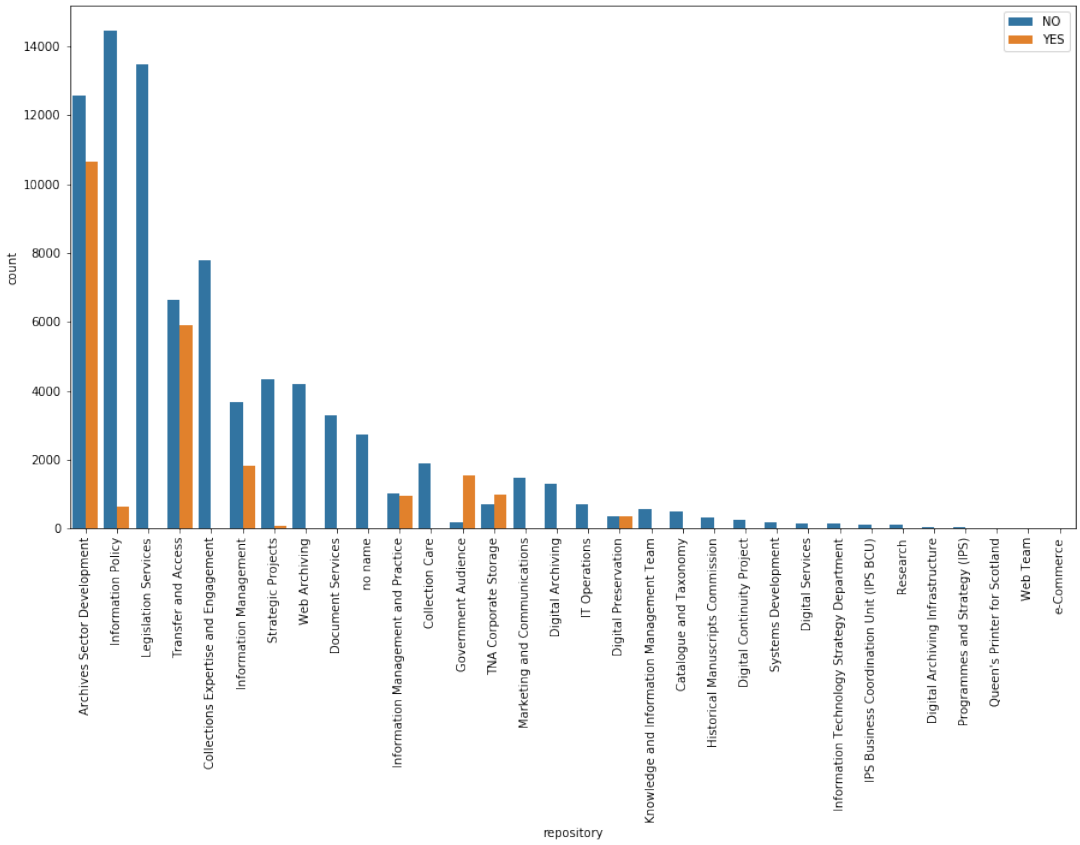

Figure 2: Distribution of documents across repositories

The Figure 4 shows volume of documents by file type. We can see that there a large number of .msg files (emails) consist of the labelled dataset. The documents are further categorised as selected (in orange) and not selected (in blue). A large portion of emails (.msg, .mbox and .MSG file types) are selected for permanent retention. Other important file types include .doc (.DOC and .docx) and .pdf and .rtf.

The document selection according to the top level folder is shown in Figure 5. We can see a correlation between the folder type to selection of files for permanent preservation. The above observations indicate the following:

- File path and top level folders are correlated with the file retention category.
- The repository (or the department) is correlated with the retention category to some extent.
- Document distribution is highly imbalanced. Categories such as 06 (permanent retention type) are have very low representation in the labelled data. Imbalance of categories can lead to misclassification errors.
- Only certain file types are highly likely to be picked up for permanent preservation. Or it is highly likely that documents have to be saved as certain file types for the sake of permanent preservation.

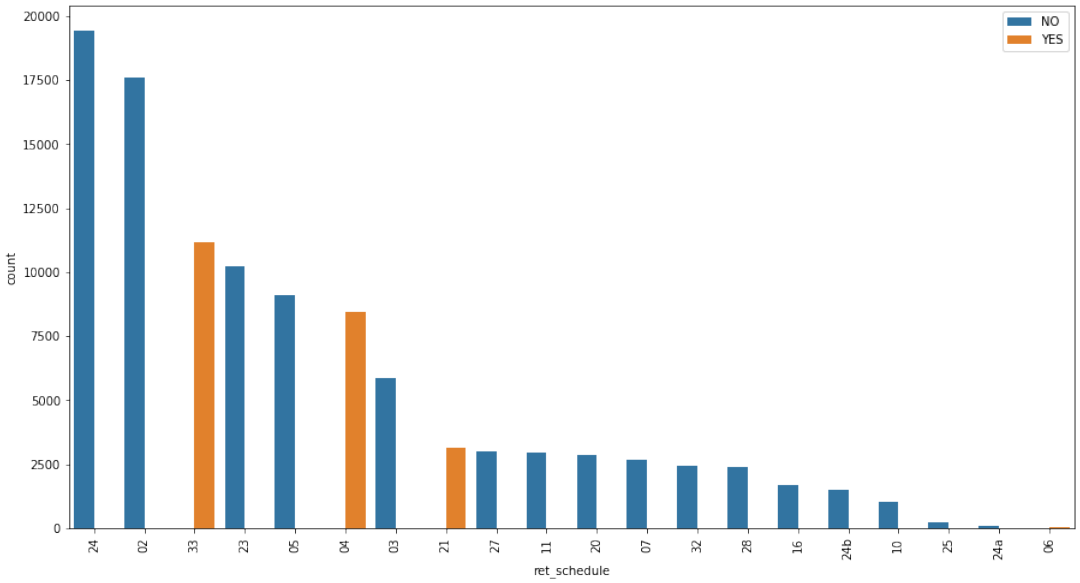

Figure 3: Distribution of documents with respect to retention schedule

3.2 Problem modelling

In order to accomplish the objective we come up with two possible approaches. (1) Explore correlation between the assigned retention schedule and document metadata. (2) Explore the contents of the document text to explore the dependence between content and retention schedule category. Both methods have their pros and cons as follows.

Exploration of metadata method is relatively easier to apply a classification model. However, it is highly likely that the model will be applicable to TNA's exemplar labelled data only. As metadata may not be available while handling data from government departments, the tool needs an extra module to extract appropriate metadata from the document data. Since this method is based on the metadata only, often it may miss sensitive documents stored in some obscure folders.

Exploration of contents of documents method requires deeper understanding of the text present in the document. We need Natural Language Processing techniques to understand the content of the data. A data pre-processing module needs to be added to the tool. An exploration of document content clusters are shown in the Figure 6. We could divide documents into six clusters for the sake of limitation of processing capability.

From the above two methods, one may conclude that given the labelled data along with metadata it may appear that the problem can be solved by applying classification models. As mentioned before, document data may not come with pre-labelled always. There may not be a supervised guidance available to us to classify documents into various categories. Problem needs to be modelled to apply unsupervised document clustering methodologies to identify the categories of documents. However, for the current problem with labelled and unlabelled datasets, we go ahead with the methodology to train a classification model on the labelled dataset and apply the model on the unlabelled data. Following section details development of prototypes by both methods.

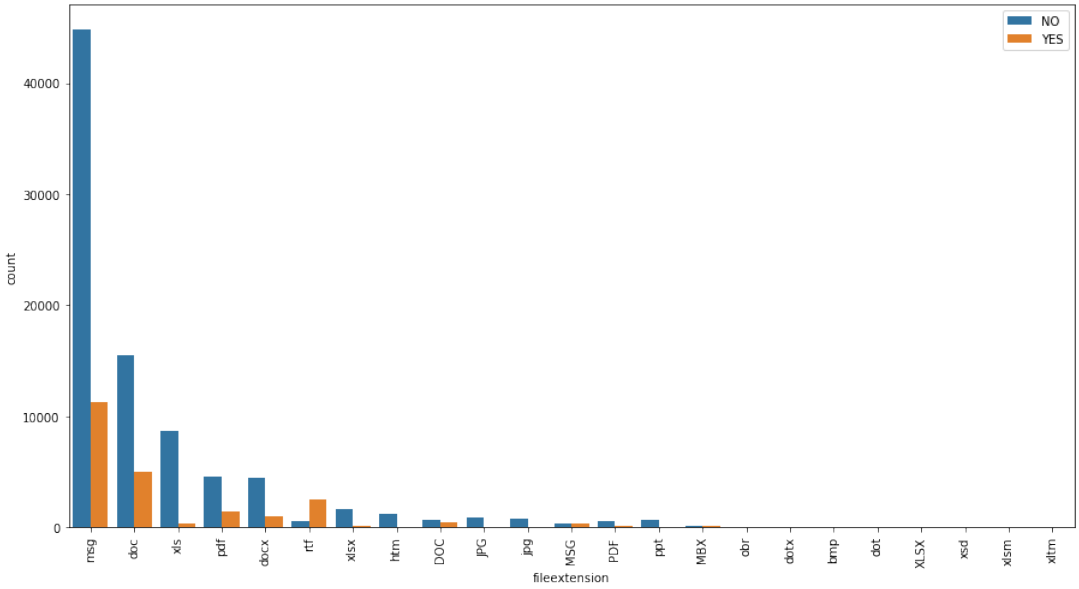

Figure 4: Volume of documents according to the file type

4 Prototype developed using metadata

The stages in the pipeline are shown in Figure 7. The labelled and unlabelled documents are loaded into the system for training and testing respectively.

Extract document features from the labelled data is to extract document metadata features such as author of the document, time last modified. However, these details were overwritten at the time of the content transferred from their original location to the hard drive provided to us. As a result, most of the metadata was corrupted and unusable.

Load metadata features were compiled by the KIM team in the form of an Excel sheet. It consist of the following features:

Feature	Description
documentid	unique id provided to each document
objectivefileid	unique id in the objective
fileextension	file format type
versionnumber	version number in objective
disposal_schedule	document retention schedule
repository	repository set for the departments
parent11 to parent1	file folder structure from 11 to 1. parent11 is the top most folder
objective2 .. 1	objective information
originalname	original name of the document
documentname	complete path to navigate in objective
trim11 to trim1	folder names trimmed to reduce to fit into excel columns

Table 1: Metadata feature Table

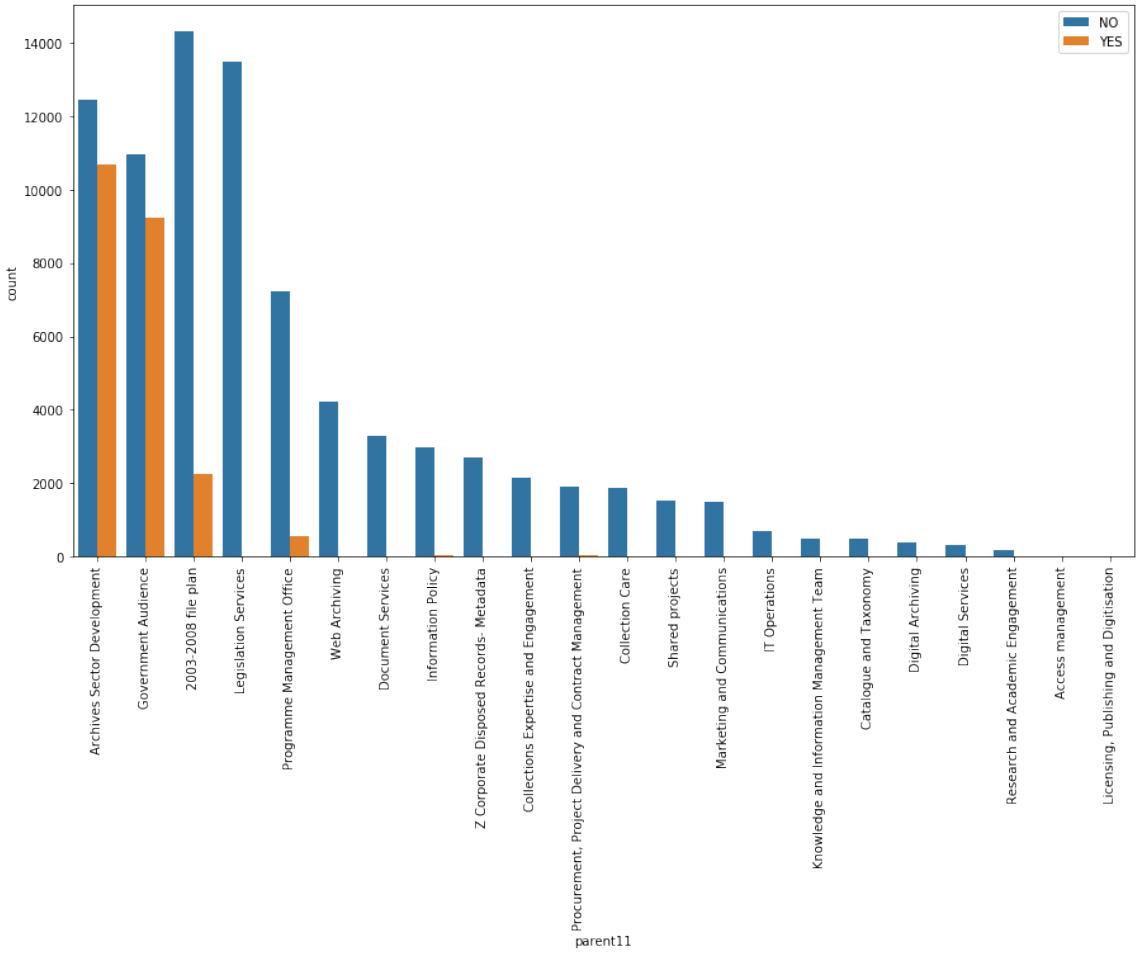

Figure 5: Distribution of documents in the folders

Feature selection: Out of the available features, only few are selected for the model development. Other redundant features are omitted. The selected features are, *file extension, disposable_schedule, repository, parent11 and document name*.

Model development on train data: Naive Bayes and Decision tree classification models were selected for experimentation to start with. Naive Bayes model is easy and fast to predict class of test data set. It performs well in multiclass prediction of retention schedules. When assumption of independence holds, a Naive Bayes classifier performs better compare to other models. However, since our train and test datasets were highly imbalanced, there is a chance that a category present in the test may not be present in the train data at all. If a category in test data set, which was not observed in training data set, then Naive Bayes model will assign a zero probability and will be unable to make a prediction. On the other side Naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously. Hence we tested the suitability of decision tree classification model.

The Decision tree models are easy to understand and interpret, perfect for visual representation. This model closely mimics the human decision-making process. Another advantage with decision tree model is it can work with numerical and categorical features with very little data pre-processing. The presence of features that depend on each other (multicollinearity) also doesn't affect the quality.

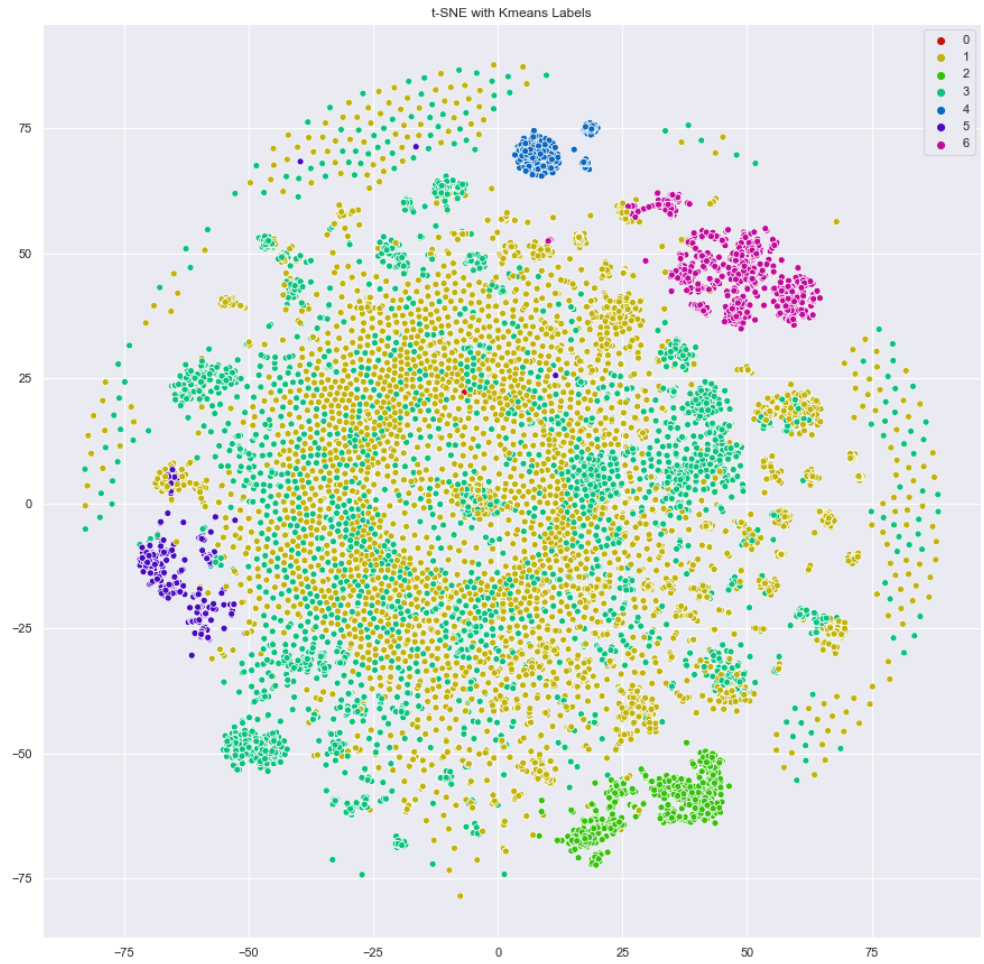

Figure 6: Document content clusters to get an insight using t-SNE

Evaluation of the Decision Tree The confusion matrix for multiclass classification by Decision tree classifier is shown in the Table 2. The Accuracy obtained by this model is 0.9324432702740953.

Model prediction on unlabelled data: When applied the above decision tree model on the unlabelled data, it classified unlabelled into three categories as follows:

5 Prototype developed using content of documents

Even though the prototype developed by the model using only metadata features, we understand that the metadata may not be available always. A model that assigns retention categories with the help of the contents lead to better results. The document clustering is done to find similar documents

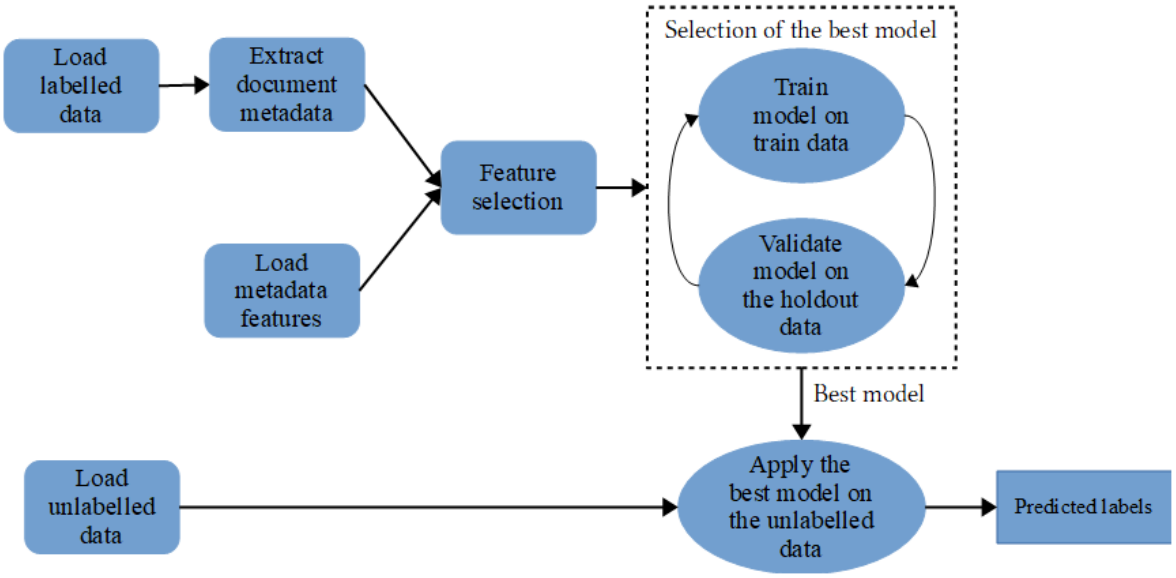

Figure 7: Complete pipeline for model development using metadata features

	02	03	04	05	06	07	10	11	16	20	21	23	24	24a	24b	25	27	28	32	33
02	3417	20	33	0	0	0	0	0	2	2	6	0	0	0	0	0	0	0	43	1
03	20	1021	40	1	0	3	0	0	0	0	11	0	4	0	0	0	0	0	33	40
04	29	54	1469	49	0	0	0	0	0	0	16	0	0	0	0	0	0	0	65	7
05	0	0	41	1706	1	5	7	5	0	5	5	6	18	1	0	0	11	0	9	1
06	0	0	0	2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07	0	0	0	1	0	499	2	0	0	0	11	19	2	0	0	0	0	0	0	4
10	0	0	0	2	0	0	201	1	0	0	2	0	0	0	0	0	0	2	0	0
11	1	0	0	9	0	0	2	515	0	0	4	4	15	0	0	0	0	40	0	0
16	4	0	0	0	0	0	0	0	320	0	2	14	0	0	0	0	0	0	0	0
20	1	0	0	3	0	0	0	0	0	533	9	0	8	0	0	0	0	1	5	17
21	7	11	6	5	0	11	1	1	1	6	555	22	0	0	0	0	0	0	0	6
23	0	0	0	3	0	15	0	9	12	1	20	1965	19	0	0	0	0	0	0	1
24	0	3	0	25	0	0	0	8	0	7	0	18	3770	0	0	0	25	3	14	15
24a	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0
24b	0	0	0	0	0	0	0	0	0	0	0	0	0	2	296	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0
27	0	0	0	13	0	0	0	0	0	0	0	0	29	0	0	0	555	0	0	0
28	0	1	0	0	0	1	0	42	0	0	0	0	2	0	0	0	0	425	4	1
32	50	33	56	8	0	0	0	0	0	8	0	0	25	0	0	0	0	4	289	11
33	1	38	4	4	0	5	0	0	0	7	4	0	15	0	0	0	0	1	7	2152

Table 2: Confusion matrix for multiclass classification by decision tree model

using the content of the document. This makes us thinking that similar documents should be grouped together. In machine learning, document clustering processes natural language to group documents together. However, for the selection of documents, the context of the content is more important than the mere similarity in linguistic procedures. In the process of document selection for permanent preservation, the false negatives are more dangerous than the false positives. A false positive raises an alarm for selecting a non-essential document. But not selecting an essential document for retention is causes a problem. The document classification process is broadly divided into four stages in the pipeline as shown in the figure 8. They are (i) data acquisition, (ii) data pre-processing and feature extraction, (iii) machine learning modelling for classification and (iv) document class prediction.

5.1 Data pre-processing and feature extraction

One of the biggest challenges with machine learning is the quality of the input data. Quite often it's not good enough. To structure 92,000+ documents, we took the help of Natural Language Processing (NLP) for data pre-processing and make data ready for the use of machine learning techniques.

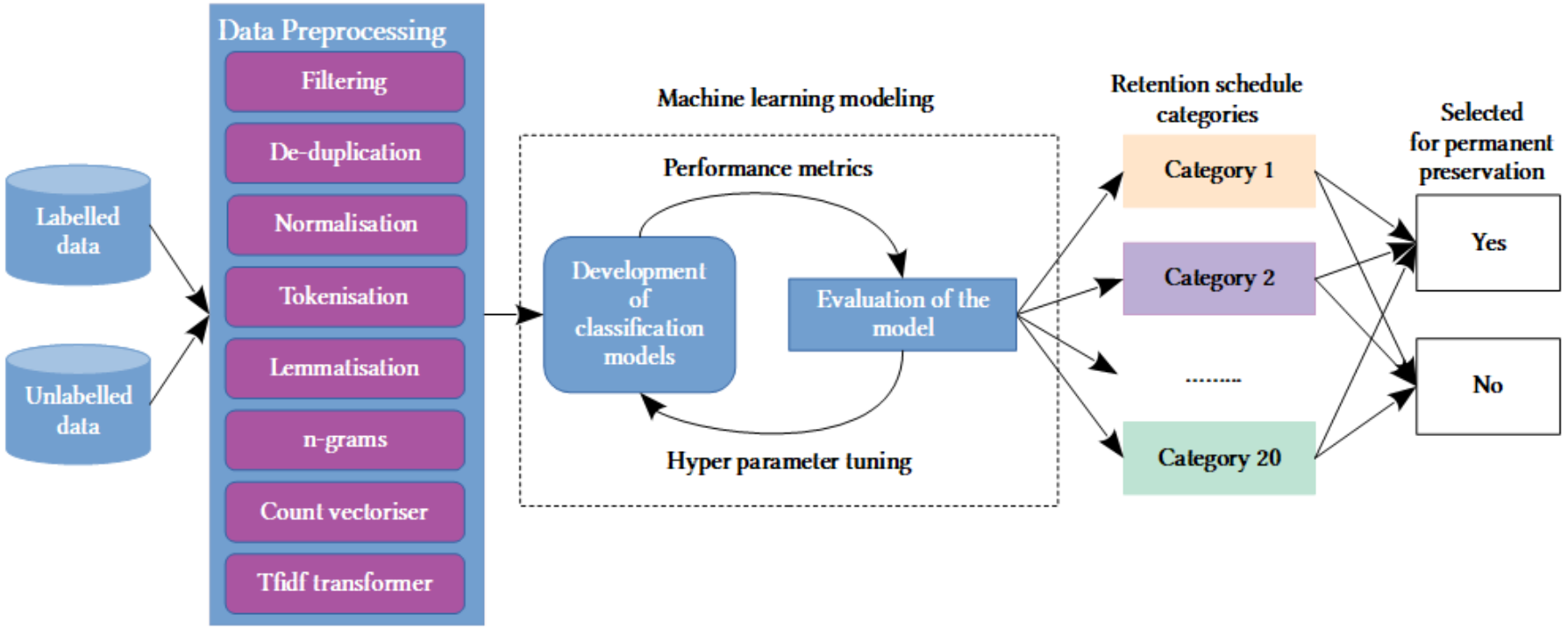

Figure 8: Document classification pipeline developed with the help of open source libraries

Retention category	Number of documents
16	930
23	49735
33	32

When designing Natural Language Processing (NLP) applications for document classification, the feature extraction becomes a significant part of the development effort. In order to develop a machine learning model, the first step is to extract features from the data. Features of the data can be thought as specific characteristics that describe the data. This step is highly recommended to describe the data as accurately as possible. However, the extraction of the feature set from the text data is not only challenging, but also complicated. The text data follows a complicated rule set defined by the language only. On the other hand, machine learning models are developed based on mathematical and statistical principles. They provide measurable and reproducible analytical outputs. Hence, we need to extract features of the text and quantify them to be fit to existing machine learning models.

The steps involved in data pre-processing of the benchmarking tool are detailed below. The data pre-processing pipeline is made as generic as possible to reuse intermediate output for KIM team to examine. Also a data scientist as an end user can refine steps to fine-tune the tool for reuse.

Document de-duplication is the process of removing duplicates from the labelled and unlabelled document set. This is to keep the documents as unique as possible. However, the class representation in our original data corpus is highly imbalanced. Many classification learning algorithms have low predictive accuracy for the infrequent class. This leads to mis-classification error for classes like '06' (shown in Figure 3). The class '06' is one of the important classes for permanent preservation. Hence the de-duplication step is removed later and the model was trained with duplicates only.

Document normalisation is the process of standardisation of text information. All documents are converted to .txt format to extract text data and metadata using Apache Tika. Document standardisation by converting to a set of text lines lead documents to lose their original structure. But, since our aim is to classify documents based on their contents and not the format, the standardisation to .txt format is acceptable.

Tokenisation is the task of chopping a sentence into small strings of characters to enable computation on text data. Several NLP libraries such as NLTK, Spacy and Gensim offer support functions to tokenise data. In the tokenisation step, our model also removes the frequently occurring stop-words such as 'is', 'was' etc.. The punctuation embedded in the input text is also cleaned during this step. The tool converts all uppercase English characters to lowercase to be compliant with standard open source libraries for computation.

Lemmatisation is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. This allows the extracted phrases to be grouped together. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger

context surrounding that sentence, such as neighboring sentences or even an entire document. We need this step to detect the document type.

n-grams is a contiguous sequence of 'n' items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs in our application. Though our tool is equipped with n-gram step, the outcome is currently not used for document classification. It will be used in our future work (document clustering).

Count Vectoriser converts a collection of text documents to a matrix of token counts and builds vocabulary from it. Our tool implements the count vectoriser from scikit-learn library. This implementation produces a sparse representation of the counts. One important use of count vectoriser is it also encodes new documents using that vocabulary. It provides a feature selection with the number of features equal to the vocabulary size found.

Tf-Idf Transformer is used on the top of count vectoriser to convert the collection of documents to a matrix of TF-IDF features. Tf-Idf transformer is used to compute the Tf-IDF scores based on the word counts computed in the previous step. the reason for not using Tf-Idf vectoriser directly is, With Tf-Idf transformer, it is possible to compute word counts and then compute the Inverse Document Frequency (IDF) values systematically and compute the Tf-Idf scores. The term count vectors can be reused for futuristic processes in document clustering.

5.2 The class imbalance problem

Data are said to suffer the class imbalance problem when the class distributions are highly imbalanced [4]. In this context, many classification learning algorithms have low predictive accuracy for the infrequent class. In our document corpus, the class imbalance is shown in the Figure 9. There are three ways to handle the class imbalance problem. They are (1) sampling, (2) algorithm approach and (3) feature selection. Sampling can be achieved by two ways, undersampling the majority class, oversampling the minority class, or by combining over and undersampling techniques. Algorithmic approach tries to optimise the performance through algorithms. For example, one-class learning methods recognized the sample belongs to that class and reject others. Under certain condition such as multi-dimensional data set one-class learning gives better performance than others [5]. The goal of feature selection approach, is to select a subset of 'k' features that allows a classifier to reach optimal performance in high dimensional datasets [5]. Since in our problem we have as many minority classes (06, 10, 16, 24a, 24b, 25, 28) as majority classes, we chose a hybrid approach of undersampling and oversampling of majority and minority classes respectively.

5.3 Classification models and Evaluation

5.3.1 Model 1 - Naive Bayes classification

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem. Naive Bayes classifiers apply Bayes' theorem with strong independence assumptions between the features [6]. Out of the many variations

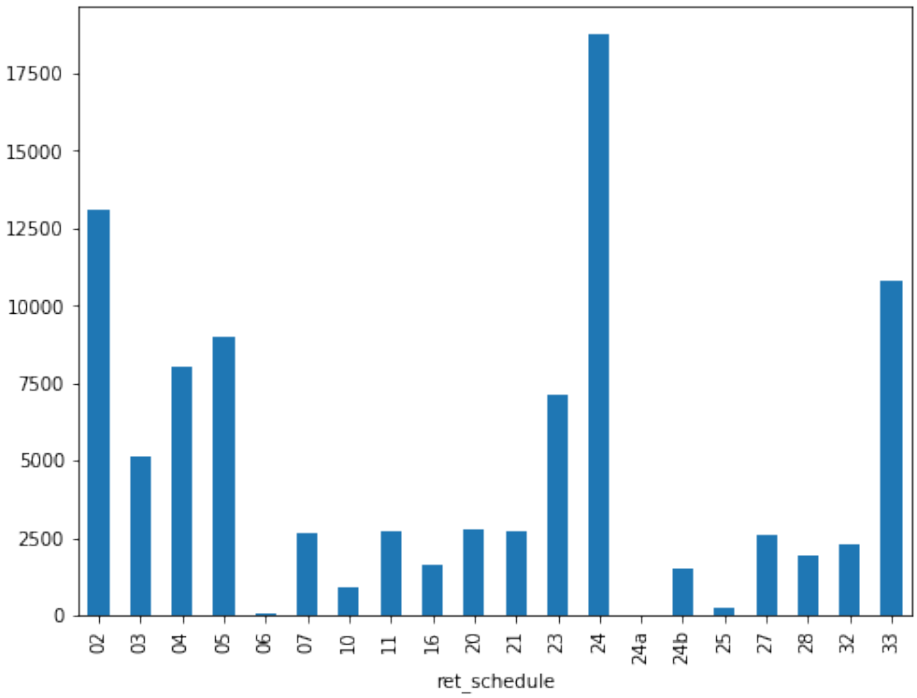

Figure 9: Documents in the document corpus according to retention schedules

of Bayes' applicability, Multinomial Naive Bayes is mostly used for document classification problem, to see whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document. We implemented the multinomial Bayes classifier from scikit learn library. Evaluation metrics for multi class and binary class are presented in tables 3 and Table 4.

Table 3: Evaluation Metrics for Naive Bayes classifier multi labelled classification

class	precision	recall	f1-score	support
02	0.77	0.62	0.69	951
03	0.81	0.93	0.87	927
04	0.91	0.85	0.88	924
05	0.84	0.71	0.77	933
06	0.88	1.00	0.93	943
07	0.78	0.81	0.80	911
10	0.88	0.83	0.85	954
11	0.64	0.83	0.72	919
16	0.94	0.98	0.96	1004
20	0.84	0.84	0.84	935
21	0.84	0.77	0.81	921
23	0.88	0.81	0.84	933
24	0.90	0.70	0.79	925
24a	0.99	1.00	1.00	933
24b	0.91	0.99	0.95	959
25	0.81	0.92	0.86	931
27	0.82	0.85	0.83	932
28	0.80	0.77	0.78	934
32	0.71	0.80	0.76	929
33	0.86	0.74	0.79	965
accuracy			0.58	18767
macro avg	0.68	0.27	0.29	18767
weighted avg	0.72	0.58	0.53	18767

Table 4: Evaluation Metrics for Naive Bayes classifier binary class classification

class	precision	recall	f1-score	support
NO	0.83	1.00	0.90	14471
YES	0.98	0.30	0.45	4296
accuracy			0.84	18767
macro avg	0.90	0.65	0.68	18767
weighted avg	0.86	0.84	0.80	18767

5.3.2 Model 2 - Logistic regression

Logistic regression is a predictive analysis method. It is used to describe the data and explain the relationship between one dependent variable and one or more independent variables. Logistic Regression is used when the dependent variable(target) is categorical. Multinomial logistic regression is a form of logistic regression used to predict a target variable have more than two classes. It is a modification of logistic regression using the softmax function instead of the sigmoid function with cross entropy loss for evaluation. The softmax function squeezes all values to the range [0,1] and the sum of the elements to 1 (one) [7]. We implemented the Logistic regression classifier from scikit learn library. Evaluation metrics for multi class and binary class are presented in ables 5 and 6.

Table 5: Evaluation Metrics for Linear classifier multi labelled classification

class	precision	recall	f1-score	support
02	0.87	0.85	0.86	951
03	0.84	0.87	0.85	927
04	0.96	0.96	0.96	924
05	0.89	0.91	0.90	933
06	0.83	0.83	0.83	943
07	0.85	0.85	0.85	911
10	0.80	0.72	0.76	954
11	0.60	0.91	0.72	919
16	1.00	0.98	0.99	1004
20	0.90	0.84	0.87	935
21	0.88	0.81	0.84	921
23	0.94	0.94	0.94	933
24	0.95	0.94	0.95	925
24a	1.00	1.00	1.00	933
24b	0.96	0.98	0.97	959
25	0.87	0.64	0.74	931
27	0.89	0.85	0.87	932
28	0.88	0.79	0.84	934
32	0.82	0.74	0.78	929
33	0.91	0.90	0.91	965
accuracy			0.89	18767
macro avg	0.88	0.86	0.86	18767
weighted avg	0.90	0.89	0.90	18767

Table 6: Evaluation Metrics for Linear Regression classifier binary class classification

class	precision	recall	f1-score	support
NO	0.97	.98	0.97	14471
YES	0.93	0.90	0.91	4296
accuracy			0.96	18767
macro avg	0.95	0.94	0.94	18767
weighted avg	0.96	0.96	0.96	18767

5.4 Discussion

The Naive Bayes assumes relative independence of words. But as expected words within a document may not be independent at all. It could reflect in over/under estimating the log likelihood. Naive Bayes did not perform well in the document classification. The recall evaluation metric is very important for the document selection problem where the false negatives are cause more problems than the false positives. However, the evaluation metric f1-score is a highly recommended one for this type of evaluation.

Conclusion

AI has been a great help to reduce burden of day-to-day work in every walk of the life. The project AI for Selection is in progress to support KIM teams in the government departments to select documents

from huge volumes of born digital documents. In this report, we have presented the benchmarking tool to evaluate commercially available products for selection problem. We have developed two approaches for the evaluation purpose. First approach uses only the metadata of the documents. This approach evaluates tools that make use of the metadata only. The second tool uses patterns and trends of the document content. It does not use any of the metadata features for model training. The second approach obtained promising results with Logistic regression classifier.

Future Work

At the moment we have assumed that data corpus comes with labelling. We used supervised learning methods to solve document classification. However, the real data may not come with labels. Some of the existing tools may be used for data labeling. However, when considering a third party tool to handle the data, we cannot sure about the correctness of labelling as it is hard for an outsider to understand the depth of context, understanding, and experience as someone inside working on the data. In this present application, the data understanding goes deeper and deeper into nuanced and technical aspects of sensitivity of policies of the government departments and archival solutions. As a future work, we need to think about a combination of unsupervised learning techniques equipped with rule-based engine.

References

[1] Transforming government through digitization. https://www.mckinsey.com/ /media/McKinsey/Industries/Public%20Sector/Our%20Insights/Transforming %20government%20through%20digitization/Transforming-government-through-digitization.ashx.

[2] Review of Government Digital Records. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/486418/Report_-_Digital_Records_Review.pdf.

[3] Charles X. Ling and Victor S. Sheng. *Class Imbalance Problem*, pages 171–171. Springer US, Boston, MA, 2010.

[4] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Napolitano. A comparative study of data sampling and cost sensitive learning. In *2008 IEEE International Conference on Data Mining Workshops*, pages 46–52, 2008.

[5] M. Wasikowski and X. Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1388–1400, 2010.

[6] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373, 2012.

[7] Scott W. Menard. *Applied logistic regression analysis*. Sage university papers series. Quantitative applications in the social sciences ; no.07-106. Sage, Thousand Oaks, Calif. ;, second edition. edition.