This is a peer-reviewed, accepted author manuscript of the following conference paper:
Venugopal, V., Ates, A., & McKiernan, P. (Accepted/In press). Machine learning for literature classification during systematic literature review – establishing the minimum threshold for labelling papers.
Paper presented at 36th Annual Conference of the British Academy of Management, Manchester, United Kingdom.

Machine learning for literature classification during systematic literature review – Establishing the minimum threshold for labelling papers

Vivek Venugopal*, Aylin Ates, Peter McKiernan

Hunter Centre for Entrepreneurship
Strathclyde Business School, University of Strathclyde
Glasgow, UK

*vivek.venugopal@strath.ac.uk | gokhan.gokmen@strath.ac.uk | akwal.sunner@strath.ac.uk

Abstract

Taking inspiration from the use of machine learning in the field of medicine for literature classification, this paper explores the use of machine learning to aid the classification of documents during systematic literature reviews in the field of business and management studies. The performances of two machine learning models, SVM and Logistic regression, are compared. The dataset used is a labelled dataset on weak signal literature. The data is iteratively split into training and testing sets with the aim of minimising the training set. The models were evaluated on Sensitivity (Recall), Precision, Specificity, Accuracy, and f1_Score to find the optimal training split. The optimal value was found to be between 40% to 50%. Which meant only 40% to 50% of the dataset needed to be labelled for the machine learning model to predict the labels for the rest of the dataset. Even though machine learning will not eliminate the labour involved in systematic literature reviews, it will save the amount of labour involved and the amount of time required.

1. Introduction

Tranfield et al. (2003), based on the literature review methods used in the field of medicine, proposed the systematic literature review (SLR) for business and management studies. They proposed the SLR for overcoming some of the shortcoming of the traditional narrative style reviews such as author's bias, non-reproducibility, coverage of papers, etc. The SLR process involves, deciding the research question/agenda, searching the database, deciding on the accept/reject criterion, classifying the papers as accept or reject based on the criterion, and finally studying the full papers from the accepted set of papers.

The procedure of SLR is indeed robust but is a very laborious and time-consuming process, both of which are in scarcity for most of the researchers. Human fatigue, is another contributor to the process which contributes to the quality of the SLR. Bannach-Brown et al. (2019) state, based on heuristic, that up to 5% misclassification error is contributed by fatigue. They also acknowledge that the effect of fatigue is under-studied and is in need of further research. With respect to the labour and the time involved Chai et al. (2021, Pg 1) state that "The screening of titles and abstracts is the most time consuming part of the review process with analysts required review thousands of articles manually, taking on average 33 days". The authors of this paper took approximately 40 – 50 days for classifying the documents.

In this paper, the final corpus of relevant papers constituted approximately only 30% of the search results. This is a very optimistic number, with typical numbers at only 2 – 3% (Ferdinands et al., 2020). The remaining 70% were not required but yet consumed precious time and energy of the authors. This was from a search result of 390 odd papers. The effect of labour and time consumed would increase linearly as the search results increase. Considering the recent developments in computing technologies such as machine learning / AI, the obvious question is why not use this technology to work for us?

In fact, machine learning (ML) methods have been actively used for literature classification in the field of medicine (For e.g., see: Bannach-Brown et al., 2019; Chai et al., 2021). But most of the studies are based on either the field of medicine or in the field of technology. The use of ML in the field of business and management (B&M) for literature classification during SLR is almost non-existent. The motivation to use ML for classification in the B&M field was based on the laborious and time-consuming experience of the authors who conducted a manual (as in human centric) SLR. And the need for drastically reducing this tedious experience not just for the authors of this paper but researchers in the B&M field in general.

Thus, the following research questions were developed for empirical research:

RQ: *Can machine learning be used to classify papers during systematic literature review in the business and management field?*

RQ: *If yes, what is the minimum number of papers that need to be labelled for training the ML model for classification?*

The above research question was broken down into parts to aid in developing the empirical research method.

RQ1: Which ML model is better suited for classification, Logistic regression, or Support Vector Machines (SVM)?

RQ2: What is the performance of the selected ML model based on various percentages of labelled training data?

To answer these questions, firstly a brief review of related work on the various ML models used for classification of papers during systematic literature is presented. This is followed by the methods section which consists of how the data was processed before it was fed to the ML model. Post this, a brief overview of the logistic regression and SVM is presented. This is then followed by the result and discussions. And finally, this paper concludes with limitations and directions for future research.

2. Related work

ML has been actively employed to classify literature in the field of medicine. Marshall and Wallace (2019) use their own ML tool called RobotSearch that reduces the number of irrelevant articles retrieved during the search in medical papers. RobotSearch is free to use but the data has to be uploaded in .RIS format. The tool is a combination of neural networks (NN) and support vector machines (SVM). Cohen et al. (2015) use a version of SVM based machine learning models to identify if a paper used randomised control trial (RCT) during literature search. The model by Cohen et al. (2015) is restricted to certain users. Both the models have been trained and validated using huge amount of data running in the thousands. "Thalia" is another online ML based semantic search engine which can detect topics rather than just key words (Soto et al., 2019).

Majority of the tools available for title and abstract based classification of papers are based on SVM and are mostly suggested for use in the medicine field. (For e.g. see:Wallace et al., 2012; Cheng et al., 2018; Yu et al., 2018; Ouzzani et al., 2016; Przybyła et al., 2018; Adeva et al., 2014). Frunza et al. (2010) use complete naive bays for text classification in systematic reviews for use in evidence-based medicine. Ferdinands et al. (2020), using various datasets within the field of medicine, compare the performance of four classification algorithms, naive bayes, logistic regression, support vector machines, and random forest. As per them naive bayes performed the best.

Table 1 shows some of the current classification tools. Most of these tools have been used in the medical and medical related fields. ASReview is an open source tool that uses a combination of various machine learning methods including deep learning (van de Schoot et al., 2021). Though ASReview's performance has been benchmarked against many literatures of diverse fields, it is yet to be benchmarked against literature in business and management field.

A common method for abstract screening is the humans 'in-the-loop' system. That is, after a typical search, the abstracts are uploaded to an ML platform and humans classify a sample of the uploaded abstracts based on the relevance Marshall and Wallace (2019). The machine then learns based on this sample and predicts the relevance of the remaining abstracts. This sort of machine learning can be called an active learning (AL) system. As Marshall and Wallace (2019, Pg 6) state, "The ideal sample size of abstracts for human classification is unknown. Exactly how many positive examples will suffice to achieve good predictive performance is an empirical question, but a conservative heuristic is about half of the retrieved set".

Table 1: Machine learning tools overview

Name of the tool	**Machine learning algorithm**
Abstrackr (Wallace et al., 2012)	SVM
ASReview (van de Schoot et al., 2021)	NB; SVM; DNN; LR and others
Colandr (Cheng et al., 2018)	SVM
FASTREAD (Yu et al., 2018)	SVM
Rayyan (Ouzzani et al., 2016)	SVM
RobotAnalyst (Przybyła et al., 2018)	SVM

Even though ASReview is based on neural networks and deep learning, it is an active learning where the sample size for training the model is left to the user's good judgement. Though many tools are available for screening literature, most of them do not mention how many labelled documents the tool needs. Some guidelines are presented but since most of them do not reveal the error rates, the actual amount of labelling to be done is yet a subjective process. Another aspect is that most of the tools have used medicine (and related fields) data.

3. Method

The authors chose do a systematic literature review on weak signals theory (Ansoff, 1975) within the strategic management and strategic foresight field of study. For this, papers were searched in two major databases, Scopus from Elsevier and Proquest ABI/inform. The search strings used were as per table 2. The search was conducted on 9th October 2021. Both databases support truncated search function which was used to get all the possible word combinations of the search string. The papers were limited to the English language and also limited to the business and management fields. The field limitation was only applied in Scopus. All sources were considered, e.g., books, journal papers, conference papers, theses & dissertation etc.

The search results from the databases were imported to the EndNote software. Using EndNote's duplicate removal function, all the duplicates were removed. The combined total of papers, hereafter called corpus, from both the databases after removal of duplicates was 528 papers. The software missed removing some duplicates due to small differences in the way the papers were stored in each of the databases. A total of 125 duplicate papers were removed. Book reviews were not considered and were dropped form the corpus. The total number of reviews removed due to this reason was 13. Making the total corpus for analysis at 390 papers. Table 3 shows the number of papers at different stages.

Table 2: Database search

Date of Search	**Database name**	**Search String**	**Number of papers**	**Limitations**
09/10/2021	Scopus	TITLE-ABS-KEY ("weak sign*" OR "seeds of change") AND (LIMIT-TO (SUBJAREA , "BUSI")) AND (LIMIT-TO (LANGUAGE , "English"))	284	Language = English, Subject Area = Business and management
09/10/2021	Proquest ABI/inform	ti("weak sign*" OR "seeds of change") OR ab("weak sign*" OR "seeds of change")	244	Source type Books, Conference Papers & Proceedings, Dissertations & Theses, Scholarly Journals

The final corpus of 390 papers was exported from EndNote as a CSV file, hereafter called data, for ease of analysis. For using a machine learning (ML) model for classification task, it needs to be first trained. The training is done on pre-labelled data. For the pre-labelled data, the authors of this paper manually classified the papers based on title and abstract. Accepted papers were labelled as 1 and rejected papers were labelled as 0. All the 390 papers were classified, and the final corpus was ready for further processing.

Table 3: Step wise corpus numbers

Initial set of papers	528
Remove duplicates	-125
Remove papers that are book reviews or where abstracts are missing	-13
Total	390

All the data manipulation, modelling, and analysis were done using the python programming language and its associated libraries. The python code was run on “Jupyter notebook” and the code is in the “.IPYNB” format. The code along with the data is available from GitHub, link to which is provided in the appendix.

3.1. Data preparation

Using the "pandas" library the data was imported as a data frame (DF). Another DF was created to contain only the "Title", "Abstract" and "Target" columns from the original DF. As the names suggests, the "Title" column consisted of the title of papers, "Abstract" consisted of the abstracts of the papers, and "Target" consisted of the classification labels (0, 1) as mentioned earlier. Any paper that did not have abstract that could be manually imputed was dropped. Two such papers were dropped bring the total number of papers to 388.

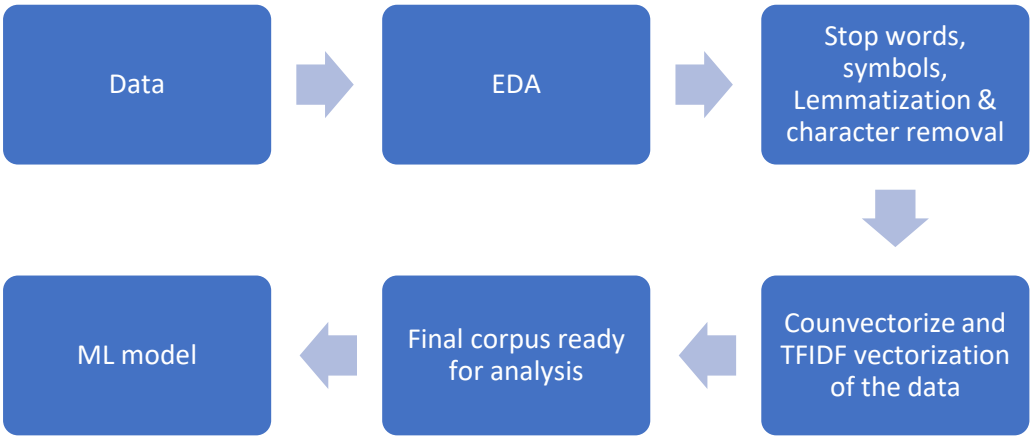

Figure 1: Data processing steps for machine learning

Figure 1 shows the schematic representation of the various steps involved in the preparation of the data before feeding it into the machine learning model.

3.2. Exploratory Data Analysis (EDA)

This paper attempts to use machine learning for classification of papers based on title and abstract. Thus, the independent variables, X (represented as a vector), were the "Title" and "Abstract" columns respectively. The dependent variable, y, was the "Target" column.

Some basic EDA was performed to understand the data on hand. The mean length based on the number of characters, inclusive of whitespaces, of titles of all papers was 83. The mean length of abstracts based on the number of characters, inclusive of whitespaces, was 1279. A detailed descriptive statistic is given in table 4.

Table 4: EDA

Statistic	Title length	Abstract length
count	388	388
mean	83	1279
std	34	658
min	11	146
25%	60	921
50%	79	1176
75%	101	1490
max	210	7926

3.3. Text processing

To be able to run an ML model on the data, it needs to be in some number form as the machine cannot understand text data. Thus, text data needs to be converted to a vector representation. But before that, the text has to be cleaned and processed. This involves, removing 1) special characters such as copyright symbols, currency symbols, html tags, etc. 2) Numerical data 3) punctuation marks 4) email addresses 5) stop words such as: if, then, it, etc. The next process would be to lemmatise the data. Lemmatisation is the process of converting of a word to its dictionary form. For e.g., the lemma form of "running" is "run".

The above set of processes is standard practice in computational text processing. The reason for applying these processes is because, if it is not done, it adds no additional value but affects the computing performance by consuming unnecessary computing power. Another important reason for the text processing is for computing the similarity measures of words which can be used to compute sentiments or in construction of chat bots etc. In this paper, the text processing is done to save computation power only. The other aspects are not applicable for this paper.

The "Title" and "Abstract" columns of the data (hereafter referred to text within this section) needed to be processed with steps as mentioned above. The text was processed by defining a function in python and applying the function on the data. The stop words were removed using the NLTK library (Bird et al., 2009) which contains a predefined list of stop words. A sample of before and after processing of the text is shown in tables 5 and 6 respectively.

Vectorising the words in the data was done using the "countvectorizer" and "tfidf" models using the "Scikit-learn" library for python (Pedregosa et al., 2011). Countvectorizer is the process of converting the data in to a "bag-of-words" representation. "Bag-of-words" is used as an analogy to communicate that the order of the words does not matter, same as items in a bag (or sack) cannot have order. The "bag-of-words" is a tabular form with values in the cell representing the number of times a term 't' is present in the document 'd', that is, the values represent the term frequencies (IIIT-B and Upgrad, 2022). The bag of words representation is shown in table 7.

Table 5: Text before processing

0	2013 International Conference of Information S...
1	An 'à la Ansoff weak signal' feedforward contr...
2	AASB 1037 sows the seeds of change: A survey o...
3	Accounting Reforms in Municipalities: The Case...
4	Actor-networking stakeholder theory for today'...
5	ADAPTIVE LEADERSHIP 101...
6	Adjacent Opportunities: Amplifying Weak Signals
7	Adjacent Opportunities: Still More Chutzpah
8	Advanced methods: Identification of promising ...
9	Advancing Chiral Chemistry in Pharmaceutical S...

Table 6: Text after processing

0	[international, conference, information, scien...
1	[ansoff, weak, signal, feedforward, control, p...
2	[aasb, sow, seed, change, survey, sgara, measu...
3	[accounting, reform, municipality, case, corpo...
4	[actornetworke, stakeholder, theory, todays, c...
5	[adaptive, leadership]
6	[adjacent, opportunity, amplify, weak, signal]
7	[adjacent, opportunity, still, chutzpah]
8	[advanced, method, identification, promise, hi...
9	[advance, chiral, chemistry, pharmaceutical, s...

Once the data is converted into a "bag-of-words" representation, the data was converted into tf-idf model. In tf-idf, tf is the term frequency and idf means the inverse document frequency. In the model, the tf-idf score is calculated for each word.

The formula for calculating tf-idf for a term 't' of a document 'd' in a document set is:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) * \text{idf}(t)$$
$$\text{and idf}(t) = \log [\, n / \text{df}(t)] + 1$$

Table 7: Bag of words

(0, 5)	1
(0, 12)	1
(0, 35)	1
(0, 36)	1
(0, 48)	1
(0, 52)	2
(0, 66)	1
(0, 86)	1
(0, 111)	1
(0, 121)	1
(0, 138)	1
(0, 144)	1
(0, 151)	2
(0, 169)	11

Where, n is the total number of documents in the document set and df(t) is the document frequency of t (Pedregosa et al., 2011). Table 8 shows an example of the tf-idf representation.

Table 8: tf-idf representation

(0, 5858)	0.03111065561768437
(0, 5833)	0.024232576829490912
(0, 5821)	0.11068646489435056
(0, 5791)	0.03111065561768437
(0, 5779)	0.03111065561768437
(0, 5777)	0.02767161622358764
(0, 5770)	0.08729494818245476
(0, 5759)	0.03111065561768437
(0, 5758)	0.005415270215494011
(0, 5751)	0.024232576829490912

Once the text was transformed using the tf-idf, this was stored as independent variable named X. The dependent variable, y, was the "Target" column as mentioned earlier. Now the data was ready to be fed into the machine learning models.

4. Machine learning models

As mentioned in the introduction, this paper aims to use the logistic regression as well as support vector machines (SVM) for aiding in the classification of papers during SLR. Specifically, the aim of this paper is to determine the minimum threshold for manual labelling of the papers before using the machine learning model for classification. For this, the data is split between training set and testing set. This splitting was done iteratively to find the optimal

training set. This optimal training value would be the minimum number of papers in the data that have to be classified manually before using ML to do the classification. To better explain the train, test split; the industry practice is to split the data as 70% to train the ML model and the remining 30% to test the performance of the ML model. During training, the test data is hidden from the ML model in order to observe how the ML predicts on new data.

To decide between logistic regression and SVM, initially the sensitivity and accuracy was determined. This was based on the initial test set based on the standard practice of splitting (70% training and 30% testing). Based on the evaluation, the chosen model was then subjected to iterative data splitting as mentioned above. The selected model was evaluated on accuracy, sensitivity (also called recall), specificity, precision, and f1_score (James, 2013) (Lumbanraja et al., 2021). The ML models were implemented using Scikit-learn library (Pedregosa et al., 2011).

Accuracy is the percentage of correctly classified data from the whole data set. This is represented as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Sensitivity is the percentage of true negatives represented as:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity is the percentage of true positives represented as:

$$Specificity = \frac{TN}{TN + FP}$$

F1-Score shows a balance between precision and recall and is represented as:

$$F1\_Score = 2 * \left(\frac{Precision * Sensitivity}{Precision + Sensitivity}\right)$$

Where, TP, TN, FP, FN are true positives, true negatives, false positives, and false negatives respectively.

4.1. Logistic regression:

Logistic regression is a classifier algorithm for categorical data based on maximum likelihood. The logistic regression uses a sigmoid function and log likelihood to find the best fit curve. The equation for the binary logistic regression is:

$$P = \frac{1}{1 + \exp -(\beta_0 + \beta_1 X)}$$

Where, β_0 is the slope intercept (decision boundary) and β_1 is the coefficient of X. The linearised expression of the binary logistic regression is given as:

$$\ln\left(\frac{P(X)}{1-P(X)}\right) = \beta_0 + \beta_1 X$$

Were, $ln\left(\frac{P(X)}{1-P(X)}\right)$ is the log likelihood, β_0 is the decision boundary (intercept), and $\beta_1 X$ is the coefficient of X.

For multinominal logistic regression the above equation changes to:

$$P = \frac{1}{1 + \exp -(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}$$

An optimal cut-off of the probability is defined for classification. This optimal cut-off is usually determined by the ROC (receiver operating characteristic) curve. ROC curve shows the performance of the model based on TP and FP at various cut-off thresholds (google.developers).

The logistic regression has a hard classification boundary. This boundary is based on the optimal cut-off point and each data point has to belong to one of the classes. For e.g., in a binary logistic model the output should belong to either yes or no (or could be 1 or 0).

4.2. Support Vector Machine (SVM)

SVM is an extension of the support vector classifier (SVC) which in turn is based on the maximal margin classifier (MMC). In the MMC, the perpendicular distance from each of the data points is calculated within a given hyperplane. The smallest distance between the observations to the hyperplane is called the margin. And the maximal margin hyperplane is where the margin is the largest (James, 2013). The data can then be classified on either side of the margins. A graphical representation can be seen in figure 2.

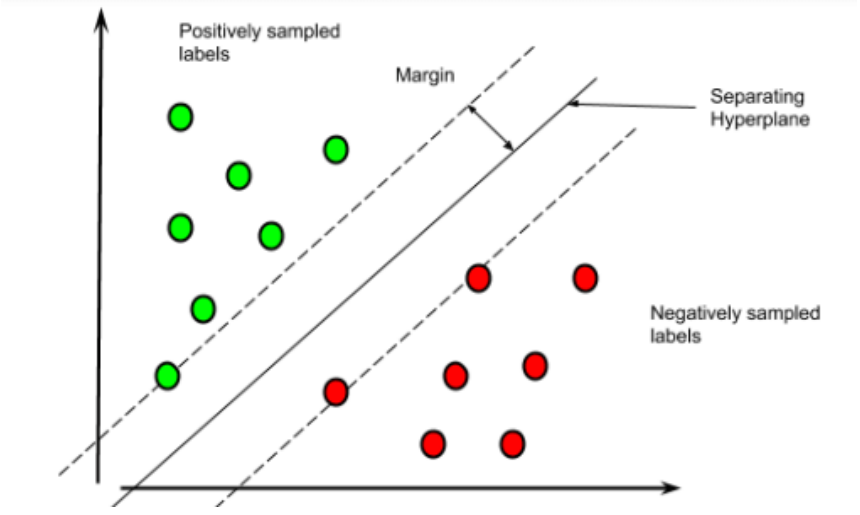

Figure 2: MMC
Source: (Chakravarti et al., 2015)

In SVC, the margin mentioned above, is considered to be soft. That is, it is acceptable to allow some of the data (observations) to violate the boundaries. The details of the support vector classifier are as follows (James, 2013) :

$$\underset{\beta_1,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{maximise} M$$

$$subject\ to \sum_{j=1}^{n} \beta_j^2 = 1,$$

$$y_i\left(\beta_0 + \beta_1 z_i 1 + \beta_2 xi2 + \cdots + \beta_p x_i p\right) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C,$$

Where, C is the nonnegative tuning parameter. M is the width of the margin which is maximised. $\epsilon_1, \ldots, \epsilon_n$ are *slack variables,* and these allow the observation to be on the wrong side of the classification. If $\epsilon_i = 0$, then that i[th] observation has not violated the margin and is on the right side of the margin. If $\epsilon_i > 0$, then that i[th] observation has violated the margin and has been misclassified. C is the sum of ϵ_i and determines the number and severity of the violations that is allowed; if $C = 0$, no violations are allowed and if $C > 0$, violations are allowed. Thus, C is considered a tuning parameter.

The observations that lie exactly on the margin are known as *support vectors* and these vectors affect the classifier. Figure 3 illustrates the support vectors and it can be seen that this figure has four support vectors.

But in practice, many a times the data does not have a linearly separable hyperplane. By using a *kernel method* this shortcoming can be overcome and this is the support vector machine.
The solution to the previous constraints of SVC can be found by (James, 2013):

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x\, , x_i \rangle$$

Where, $\langle x, x_i \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}$ which is nothing but the inner product of the observations. α_i and β_0 are the training parameters that need to be estimated. And α_i is nonzero for the support vectors. The inner product form can be generalised for applying kernel as:

$$K(x_i, x_i')$$

One of the popular kernel choices is the *radial kernel,* which is as shown below (James, 2013):

$$K\ (x_i,\ x_{i'})\ =\ \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij}\ -\ x_{i'j})^2\right)$$

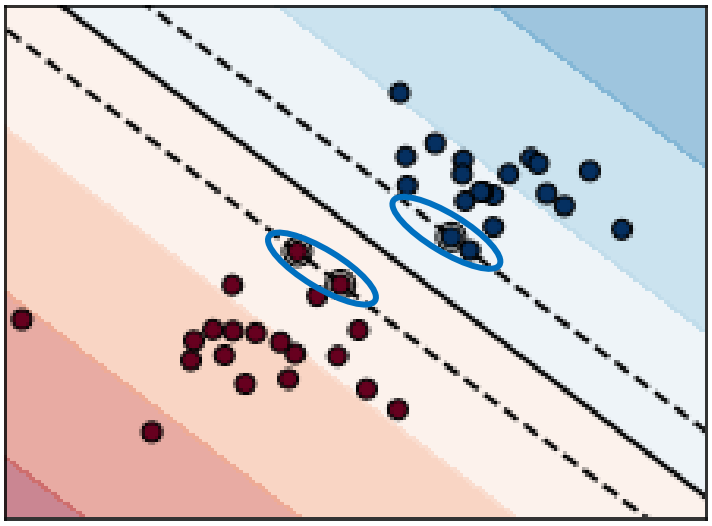

Figure 3: SVC
Source: (Pedregosa et al., 2011)

Where, $-\gamma$ is a positive constant, similar to α_i in the equation mentioned earlier, and $\left(x_{ij} - x_{i'j}\right)^2$ is the *Euclidean distance.*

If x^* is the test observation and if x_i the train observation; Euclidean distance would become large if x^* and x_i are far apart and therefore the exponent of this would become very tiny. And this x_i will have no influence on $f(x)$. Thus, only those training and testing observations which are close to each other will have an influence on $f(x)$. This is especially useful when the data does not have a linearly separable hyperplane.

Figure 4 shows the radial kernel SVM based classification of a non-linearly separable data.

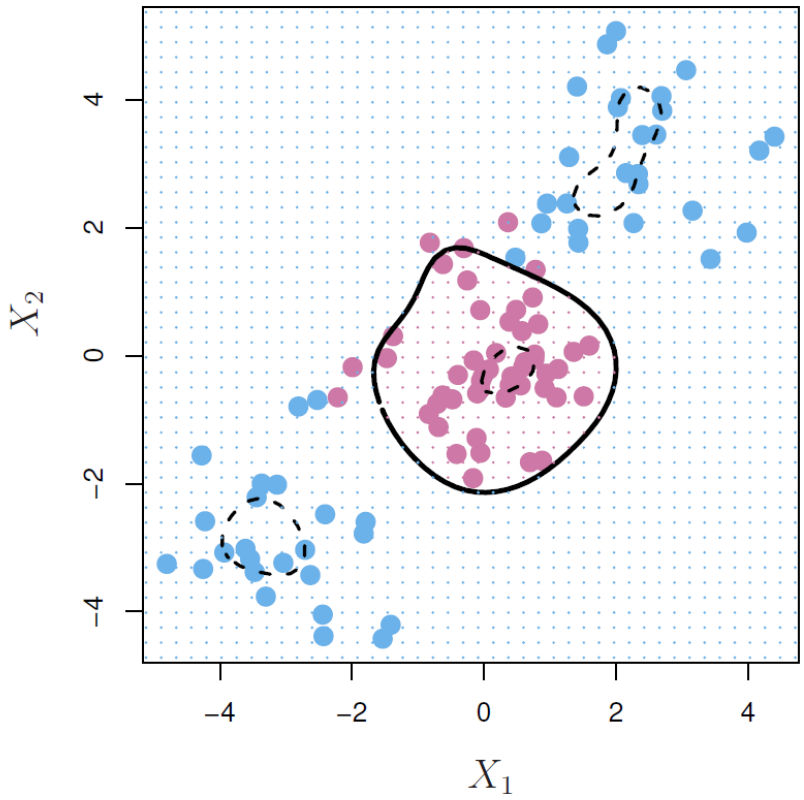

Figure 4: Radial kernel SVM
Source: (James, 2013)

5. Results and Discussion

The logistic regression's performance even with the standard practice of train, test split was poor. The accuracy of the model on test data was 79% and sensitivity (recall) was 60%. Since the aim is to reduce the training set to enable less manual work for classification during SLR, it was decided to drop logistic regression model based on the evaluation. As the performance would only come down or, more optimistically, remain the same, when the training set is reduced.

The SVM performed well with respect to the standard test, train split of 70% and 30%. The accuracy of the model on train data was 80% and the sensitivity was 99%. This was a very promising result. Thus, it was decided to use SVM for iteratively evaluating the model performance for different training sets. Table 9 shows the values of the results. The iterations began from considering 10% as training data and ended at 90% as training data.

Table 9: Results

Train_value	Sensitivity (Recall)	Precision	Specificity	Accuracy	f1_Score
0.1	0		1	0.562857	
0.2	0.149254	1	1	0.633441	0.25974
0.3	0.461538	0.931034	0.974194	0.753676	0.617143
0.4	0.602041	0.907692	0.955556	0.806867	0.723926
0.5	0.646341	0.883333	0.9375	0.814433	0.746479
0.6	0.608696	0.954545	0.977011	0.814103	0.743363
0.7	0.622642	0.916667	0.953125	0.803419	0.741573
0.8	0.628571	0.916667	0.953488	0.807692	0.745763
0.9	0.5	0.727273	0.869565	0.717949	0.592593

The aim of this paper was to determine the minimum amount of manually labelled data required for training a machine learning model to classify papers. This way both manual labour and time can be saved. Choosing this training value is a trade-off between model performance and the reduction of labour. If the highest evaluation is chosen, then more labour is required.

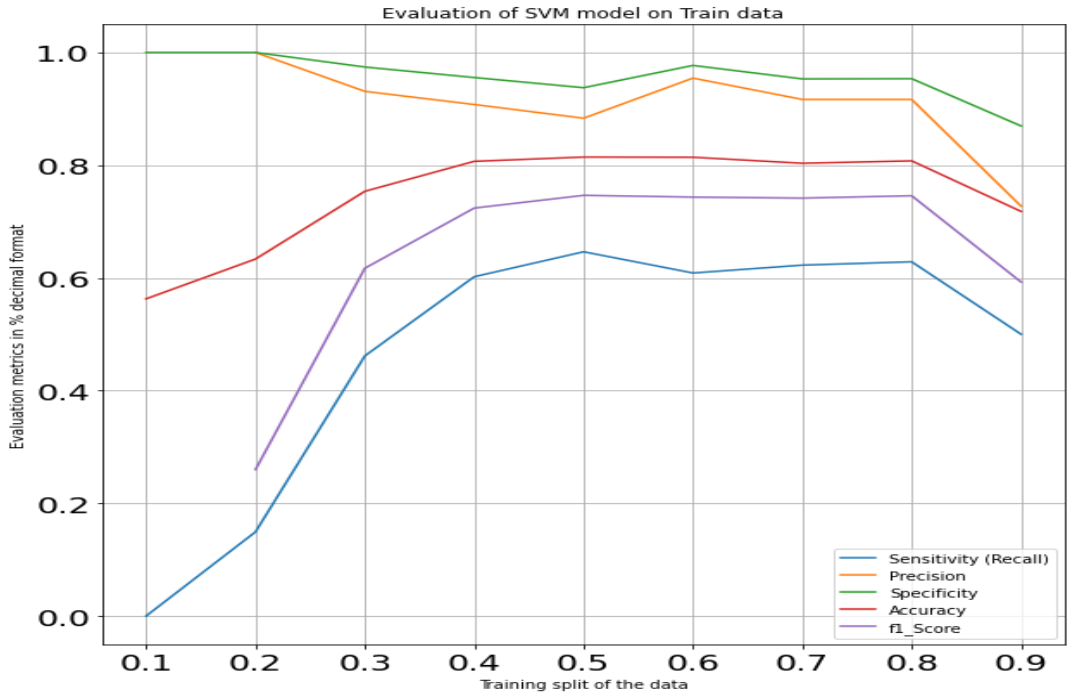

Figure 5: Evaluation metrics plot

In table 9 a blue box is highlighted to show the suggested training value considering all the evaluation parameters. This is between 40% (0.4) to 50% (0.5) which would mean manually labelling 156 to 195 papers. This would save quite an amount of labour and time compared to labelling 390 papers.

Figure 5 shows the graphical representation of table 9. Train_value in table 9 represents the training set as a percentage of the whole corpus. For e.g., train_value of 0.20 would mean 20% of the corpus is chosen to be the training data for the machine to learn. From table 9 and figure 5, it can be observed that the specificity and accuracy is dropping as the train_value is increasing. At train_value of 0.5 (50%) the specificity and accuracy are 93% and 81% respectively. The accuracy starts to drop from 81% all the way till 71% after the train_value of 0.5. The two most important evaluation parameters are sensitivity and specificity as they represent the model's capability to correctly identify the negatives and positives (0s and 1s in this case). The value of specificity increases to 97% and the starts to drop to 95% and then drops further to 86% at train_value of 0.90 (90%). Sensitivity is the highest at train_value of 0.50, then drops and again raises a bit before dropping again. The F1 score is hovering at 73% and 74% after train_value 0.40. This shows that there is a good balance between precision and sensitivity.

Considering all the evaluation parameters and considering the aim of the paper, it is suggested to keep the manual labelling between 40 – 50% of the corpus. This is in line with the present industry thumb rule (Marshall and Wallace, 2019). Choosing the train_value is an individual choice. But for this paper the choice is between 40 – 50% and choosing amongst this is again an individual choice. Specificity and precision are higher at train_value of 40% compared to train_value of 50%. Whereas, at train_value of 40%, accuracy and sensitivity are lower than train_value of 50%.

Since the priority during SLR is to correctly identify the acceptable papers, specificity is an important evaluation parameter. Thus, manually labelling 156 papers would have been sufficient with respect to this data.

6. Limitations and future research

As with any research, this paper is not free from limitations. These limitations can be taken up as future research. The foremost limitations are with respect to the data. More variety of data from various fields within the management and business should be considered. Not just diverse data but the SVM model's performance also needs to be subjected to various data sizes to make the results of this paper more generalisable.

The second limitation is with respect to the parameter tuning of the model. The regularisation parameter "C" was not tuned and only the default parameter of the sklearn library was used. Only the 'rbf' kernel, which is the default kernel within the Sklearn library for SVM was used. The performance of SVM with other kernel methods needs to be investigated. Tuning the other optimising parameters could further improve the model.

The final limitation is with respect to the machine learning methods. Only SVM and logistic regression were considered. Other simpler models such as a naive bays or more complex models such as neural networks and deep learning need to be considered.

References

Adeva, J. J., Pikatza, J., Carrillo, M. & Zengotitabengoa, E. 2014. Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications: An International Journal,* 41**,** 1498-1508.

Ansoff, H. I. 1975. Managing strategic surprise by response to weak signals. *California Management Review,* 18**,** 21.

Bannach-Brown, A., Przybyła, P., Thomas, J., Rice, A. S. C., Ananiadou, S., Liao, J. & Macleod, M. R. 2019. Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error. *Systematic Reviews,* 8**,** 23.

Bird, S., Klein, E. & Loper, E. 2009. *Natural language processing with Python*, Beijing, Cambridge Mass. : O'Reilly.

Chai, K. E. K., Lines, R. L. J., Gucciardi, D. F. & Ng, L. 2021. Research Screener: a machine learning tool to semi-automate abstract screening for systematic reviews. *Systematic Reviews,* 10**,** 93.

Chakravarti, M., Kothari, T. & Rajput, M. 2015. A Comprehensive Study On The Applications Of Machine Learning For Diagnosis Of Cancer.

Cheng, S. H., Augustin, C., Bethel, A., Gill, D., Anzaroot, S., Brun, J., DeWilde, B., Minnich, R. C., Garside, R., Masuda, Y. J., Miller, D. C., Wilkie, D., Wongbusarakum, S. & McKinnon, M. C. 2018. Using machine learning to advance synthesis and use of conservation and environmental evidence. *Conservation Biology,* 32**,** 762-764.

Cohen, A. M., Smalheiser, N. R., McDonagh, M. S., Yu, C., Adams, C. E., Davis, J. M. & Yu, P. S. 2015. Automated confidence ranked classification of randomized controlled trial articles: an aid to evidence-based medicine. *Journal of the American Medical Informatics Association,* 22**,** 707-717.

Ferdinands, G., Schram, R., de Bruin, J., Bagheri, A., Oberski, D., Tummers, L. & Schoot, R. 2020. *Active learning for screening prioritization in systematic reviews - A simulation study*.

Frunza, O., Inkpen, D. & Matwin, S. 2010. *Building Systematic Reviews Using Automatic Text Classification Techniques*.

google.developers. *Macine Learning Crash Course* [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=An%20ROC%20curve%20(receiver%20operating,False%20Positive%20Rate [Accessed 04/03/2022 2022].

IIIT-B & Upgrad 2022. Lecture notes in lexical processing.

James, G. 2013. *An introduction to statistical learning : with applications in R / [internet resource]*, New York, NY : Springer.

Lumbanraja, F. R., Fitri, E., Junaidi, A. & Prabowo, R. Abstract Classification Using Support Vector Machine Algorithm (Case Study: Abstract in a Computer Science Journal). Journal of Physics: Conference Series, 2021. IOP Publishing, 012042.

Marshall, I. J. & Wallace, B. C. 2019. Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *SYST REV-LONDON,* 8**,** 163-10.

Ouzzani, M., Hammady, H., Fedorowicz, Z. & Elmagarmid, A. 2016. Rayyan—a web and mobile app for systematic reviews. *Systematic Reviews,* 5**,** 210.

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.

Przybyła, P., Brockmeier, A. J., Kontonatsios, G., Le Pogam, M.-A., McNaught, J., von Elm, E., Nolan, K. & Ananiadou, S. 2018. Prioritising references for systematic reviews with RobotAnalyst: A user study. *Research Synthesis Methods,* 9**,** 470-488.

Soto, A. J., Przybyła, P. & Ananiadou, S. 2019. Thalia: semantic search engine for biomedical abstracts. *Bioinformatics,* 35**,** 1799-1801.

Tranfield, D., Denyer, D. & Smart, P. 2003. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management,* 14**,** 207-222.

van de Schoot, R., de Bruin, J., Schram, R., Zahedi, P., de Boer, J., Weijdema, F., Kramer, B., Huijts, M., Hoogerwerf, M., Ferdinands, G., Harkema, A., Willemsen, J., Ma, Y., Fang, Q., Hindriks, S., Tummers, L. & Oberski, D. L. 2021. An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence,* 3**,** 125-133.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J. & Trikalinos, T. A. 2012. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium.* Miami, Florida, USA: Association for Computing Machinery.

Yu, Z., Kraft, N. A. & Menzies, T. 2018. Finding better active learners for faster literature reviews. *Empirical Software Engineering,* 23**,** 3161-3186.