Determining Receipt Validity From E-mail Subject Line Using Feature Extraction And Binary Classifiers

Chanda Hirway [1, 2], Enda Fallon [1], Paul Conolly [2], Kieran Flanagan [2], Deepak Yadav [2]

[1] Software Research Institute, Athlone Institute of Technology, Athlone, Ireland
chanda.hirway@ait.ie, efallon@ait.ie

[2] The NPD Group, L.P., Athlone, Ireland
paul.connolly@npd.com; kieran.flanagan@npd.com; deepak.yadav@npd.com

***Abstract* – Many data quality technologies are available to manage diverse types of data as the number of structured and unstructured data sources grows. Modern data quality solutions can improve efficiency and decrease risks by compensating for missing or erroneous data before it is stored in the data warehouse. To improve the accuracy of data processing models, data quality solutions employ machine learning and natural language processing capabilities. The purpose of this study is to determine the authenticity of a customer invoice based on the subject line of an email using feature extraction and binary classifiers. A Bag of Words (BOW) feature extraction method is implemented to create a vocabulary and count its frequency. To determine the accuracy of a receipt from the subject line of received emails, three binary classifiers were used: Naive Bayes Bernoulli NB, Support Vector Machine, and Random Forest. Furthermore, these three classifiers were compared based on their accuracy, precision, recall, and F1 score. To determine True Positive, True Negative, False Positive, and False Negative values, a Confusion Matrix was built. The Random Forest classifier was found to be more effective in terms of accuracy, precision, recall, and F1 score. Other classifiers must also be incorporated to further reduce the False Negatives values, which play a significant part in calculating model accuracy.**

Keywords – Feature Extraction, Bag of Words, TF-IDF, Naïve Bayes Bernoulli NB, Support Vector Machine, Random Forest Confusion Matrix, Accuracy

I. INTRODUCTION

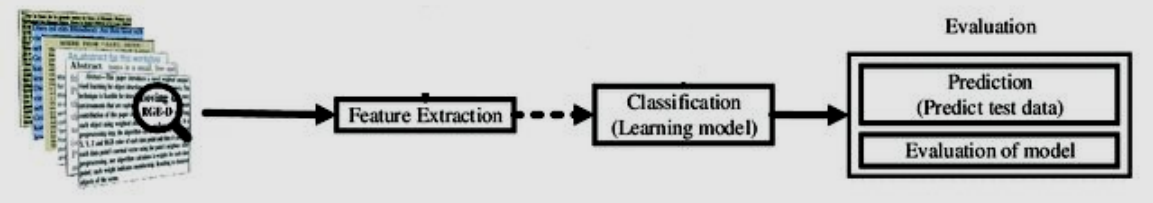
Data analytics has the ability to transform every element of a retail operation by giving vital information to departments. The process through which merchants locate, acquire, and interpret data created across departments is known as retail analytics. These strategies produce a set of actionable insights based on company trends, evolving patterns, and performance data. These data could be utilized to improve critical parts of the retail industry, including customer service and sales. Data analytics helps companies become more competitive by supporting them in making data-driven decisions.

Firms must issue a significant number of invoices every day. Each invoice contains essential information such as the purchase date and time, the products purchased and their costs, and payment details. This task can be quite labor-intensive and manual. The categorization, which is also the topic of this research, employs machine learning algorithms capable of extracting relevant knowledge from a set of received email messages to determine the validity of a receipt by analyzing email subject line.

Section II discuss about the pipeline of the text classification. Section III of the paper details related work in the area of text classification. Section IV discusses the most widely used text preprocessing techniques and classification techniques. Section V comprises of the experimental design and the methodology undertaken for the project. Section VI details all the results that were observed after conduction the various experiments listed in section V and section VII includes the conclusion and the proposed future work.

II. PIPELINE OF TEXT CLASSIFICATION

Over the last few decades, text categorization problems have been extensively researched and solved in a variety of real-world applications [1]. The following phases can be deconstructed in most text classification and document categorization systems: Extraction of features, classifier selection, and evaluations. The structure and technical implementations of text classification systems are discussed in this work in terms of the pipeline is shown in Figure 1.



-Figure 1: Pipeline of Text Classification

The initial pipeline input consists of some raw text data set. In general, text data sets contain sequences of text in documents as $D=\{X_1,X_2,\ldots,X_N\}$ where Xi refers to a data point (i.e., document, text segment) with *s* number of sentences such that each sentence includes w_s words with l_w letters. Each point is labeled with a class value from a set of *k* different discrete value indices [1].Then, for training purposes, a structured set called Features Extraction is developed. The most important stage in document categorization is selecting the best classification method. The evaluation step, which is separated into two parts, is the other half of the pipeline (prediction the test set and evaluating the model).

A. Feature Extraction:

Texts and documents, in general, are unstructured data sets. When employing mathematical modeling as part of a classifier, these unstructured text sequences must be translated into a structured feature space. First, the data must be cleaned to remove any extraneous characters or words. Formal feature extraction approaches can be used once the data has been cleansed. Bag of Words (BOW), Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF) [2], Word2Vec [2], and Global Vectors for Word Representation (GloVe) [3] are common feature extraction techniques.

B. Classification Techniques:

Choosing the best classifier is the most critical stage in the text classification pipeline. We cannot successfully determine the most efficient model for a text categorization application until we have a complete conceptual knowledge of each algorithm. The NBC (Naive Bayes Classifier) was a widely used classification machine learning algorithm. We'll take a quick look at the Nave Bayes Classifier, which is both computationally and memory-efficient [5]. Another prominent technique that uses a discriminative classifier for document categorization is Support Vector Machine (SVM) [6,7]. This technique can also be applied in other areas of data mining, such as bioinformatics, image and video processing, human activity classification, safety and security, and so on. This model is also utilized as a benchmark for many scholars to measure their own work against in order to highlight novelty and contributions. Document categorization has also been examined using tree-based classifiers such as decision tree and random forest [8].

C. Evaluation:

The final step in the text categorization pipeline is evaluation. Understanding how a model operates is critical for the use and development of text classification systems. There are numerous methods for evaluating supervised techniques. The simplest way of evaluation is accuracy calculation [10].

III. LITERATURE REVIEW AND LIMITATIONS OF EXISTING TECHNIQUES

Text can be a very rich source of information, but due to its unstructured nature, extracting insights from it can be difficult and time-consuming. It is estimated that approximately 80% of all information is unstructured, with text being one of the most prevalent types of unstructured data. Because text is messy, analyzing, understanding, organizing, and sorting through text data is difficult and time-consuming.

Text classification can be done in two ways: manually or automatically. A human annotator interprets the contents of the text and categorizes it properly in manual text classification. This procedure can produce excellent results, but it is time-consuming and costly.

Machine learning, Natural Language Processing (NLP), and other AI-guided approaches are used to automatically identify text in a faster, more cost-effective, and more accurate manner using automatic text classification. Although there are numerous ways to automatic text classification, they all fall into one of three categories:

- Rule-based systems
- Machine learning-based systems
- Hybrid systems

A. Rule - based systems

Rule-based (RB) techniques use a collection of customized language rules to classify text into ordered categories. These rules enable the system to find suitable categories based on the content of a text by using semantically relevant components of the text. An antecedent or pattern and a projected category constitute up each rule.

Limitations:

i. A lot of manual labor: The RB system necessitates extensive domain knowledge as well as a significant amount of manual labor.

ii. Time-consuming: Creating rules for a complex system is difficult and time-consuming.

Rule-based systems are also difficult to maintain and do not scale well because adding new rules can affect the outcomes of existing rules. This is where machine learning for text classification can provide an advantage.

B. Machine Learning – based systems

Machine learning text classification learns to make classifications based on past observations rather than relying on manually crafted rules. Machine learning algorithms can learn the various associations between pieces of text and that a specific output (i.e., tags) is expected for a specific input by using pre-labeled examples as training data (i.e., text). A "tag" is a pre-determined classification or category into which any given text may fall.

Machine learning text classification is typically much more accurate than human-crafted rule systems, especially on complex NLP classification tasks. Machine learning classifiers are easier to maintain, and you can always tag new examples to learn new tasks.

The existing system used a database to store the subject lines of customer invoice emails along with other fields such as receipt validity. A rule-based approach was used to classify whether the email was a valid receipt based on keyword/s present in the subject line and accordingly set the receipt validity flag. A lookup table with stored keywords was used for this rule-based classification. Considering the limitations of the rule-based system, the proposed system uses the Machine Learning approach of text classification to validate receipts.

IV. PROPOSED MACHINE LEARNING TECHNIQUES

A. Text Cleaning and Pre-processing

For text classification applications, feature extraction and pre-processing are critical tasks. We describe methods for cleaning text data sets in this section, which removes implicit noise and allows for informative featurization.

Most text and document data sets contain a large number of unwanted words, such as stopwords, misspellings, slang, and so on. Noise and superfluous features can have a negative impact on system performance in many algorithms, particularly statistical and probabilistic learning algorithms. In this section, we'll go through several text cleaning and pre-processing approaches and methodologies.

A1. Tokenization: Tokenization is a pre-processing technique that divides a stream of text into tokens, which are words, phrases, symbols, or other significant pieces [13,14]. The investigation of the words in a sentence [40] is the primary purpose of this step.

A2. Stopwords: Many words commonly employed in text and document categorization algorithms have no significant meaning, such as "a", "around", "above", "across", "after", "afterwards", "again", and so on. Taking these words out of texts and documents is the most usual method of dealing with them [15].

A3. Stemming: In NLP, a single word might exist in multiple forms (for example, singular and plural noun forms), all of which have the same semantic meaning [16]. Stemming is one way for combining different variants of a word into the same feature space. For example, the stem of the word “walking” is “walk”.

A4. Lemmatization: Lemmatization is a natural language processing technique that replaces a word's suffix with a different one or removes the suffix entirely to obtain the fundamental word form (lemma) [17, 18, 19].

B. Weighted Words:

The most basic type of weighted word feature extraction is Term Frequency (TF), in which each word is mapped to a number that represents the number of times it appears in the corpus. Word frequency is typically used as a boolean or logarithmically scaled weighting in methods that extend the results of TF. Each document is translated into a vector (with the same length as the content) holding the frequency of the words in that document in all weight words techniques.

B1. Bag of Words (BoW): The bag-of-words model (BoW model) is a reduced and simplified representation of a text document based on specified criteria such as word frequency. The BoW technique is used in a variety of fields, including computer vision, natural language processing, Bayesian spam filters, document categorization, and information retrieval using Machine Learning. A body of text, such as a document or a sentence, is viewed as a collection of words in a BoW. In the BoW procedure, lists of words are formed. The semantic relationship between these words is neglected in the collecting and construction of these words in a matrix, which are not sentences that structure sentences and grammar. The content of a sentence is frequently represented by the words. While grammar and appearance order are ignored.

Limitation of Bag of Words (BoW): Bag-of-words models encode each word in the vocabulary as a single-hot-encoded vector, for example, each word is represented by a

dimensional sparse vector with 1 at the index corresponding to the word and 0 at all other indexes for a vocabulary. Bag-of-word models have scalability issues as vocabulary grows into the millions (for example, "This is good" and "Is this good" have the same vector representation). The bag-of-word technical difficulty is also a major challenge for the computer science and data science communities.

C. Classification Algorithms

Two common ensemble learning algorithm techniques: boosting and bagging were discussed. Some methods, such as logistic regression, Naive Bayes, and k-nearest neighbor, are more traditional, yet they are still widely employed in science. Support vector machines (SVMs), particularly kernel SVMs, are widely utilized as classification algorithms. For document categorization, tree-based classification algorithms such as decision trees and random forests are fast and accurate.

C1. Boosting and Bagging: For document and text data set classification, voting classification approaches such as bagging and boosting have been successfully developed [20]. Boosting adjusts the distribution of the training set dependent on the performance of prior classifiers, whereas bagging does not [21].

C2. Naïve Bayes Classifier: Naive Bayes text classification has been frequently utilized for document categorization. The Bayes theorem, which was developed by Thomas Bayes, is the theoretical foundation of the Naive Bayes classifier method. Recent research has focused on this strategy in the context of information retrieval [22]. This method is a generative model, which is the most common way of categorizing text. The most basic version of NBC was created by bag-of-word, a feature extraction technique that counts the quantity of words in documents.

Limitation of Naïve Bayes Algorithm: The Naive Bayes algorithm has a number of drawbacks. NBC makes a bold assumption regarding the data distribution's structure [23, 24].

C3. Support Vector Machine: SVM was created to do binary classification jobs. However, many scholars use this dominate strategy to solve multi-class issues [25].

Binary-Class SVM: In the context of text classification, let $x_1, x_2, \ldots, x_l$ be training examples belonging to one class X, where X is a compact subset of R^N [26]. Then we can formulate a binary classifier as shown in equation 1,2 and 3 below:

$$\min \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - p \qquad (1)$$

Subject to:

$$(w.\phi(x_i)) \geq p - \xi_i \; i = 1,2,\ldots\ldots,l, \xi \geq 0 \qquad (2)$$

If w and p solve this problem, then the decision function is given by:

$$f(x) = sign((w.\phi_{(x)}) - p \qquad (3)$$

$$f(x) = sign((w.\Phi(x)) - p)$$

The main limitation of SVM when applied to string sequence classification is time complexity. However, the lack of transparency in results caused by a large number of dimensions limits the SVM algorithms for text categorization.

C4. Random Forest: The approach of random forests, sometimes known as random choice forests, is an ensemble learning method for text classification. The primary concept of RF is to create random decision trees.

Limitations of Random Forest: In comparison to other techniques such as deep learning, random forests (i.e., ensembles of decision trees) are very fast to train for text data sets, but they are slow to make predictions once trained [27].

V. EXPERIMENTAL DESIGN

The data was extracted from the real-world consumer Email messages from the database and 2000 samples of data were used for the experiment. The entire dataset was divided into training (D_{tr}) and testing data (D_t) in the ratio of 70:30. The dataset description in Table I consists of the dataset name, its description, and the number of messages in the dataset while the classes to be detected are detailed in Table II.

TABLE I. DATASET DESCRIPTION

Dataset	Description	Size
D_{tr}	Training Dataset	1400 messages
D_t	Test Dataset	600 messages

TABLE II. CLASS DESCRIPTION

No	Class	Description
1	0	Represent the class for a valid receipt
2	1	Represent the class for an invalid receipt

The algorithms were implemented using Python 3.8. The scikit-learn library of Python was used for general purpose machine learning. NLTK library was used for Natural Language Processing.

A. Methodology:

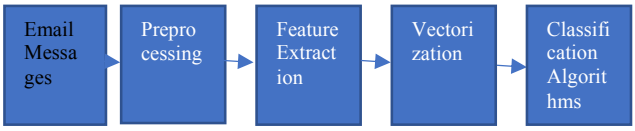


Figure 2: Flow Diagram of the project

A1. Input: The first task in every text processing task is to read the data. There were 2000 email messages which were stored in rows and 25 columns in the dataset. The unwanted columns were removed, and the final dataset consisting of 2000 rows and 2 columns were used for further processing of data. The total number of emails in the dataset were 2000, out of which 558 were valid receipt and 1442 were invalid receipt.

Given a collection of training documents $D_{tr} \subset D$ labelled as a receipt (1) and non-receipt (0), these algorithms use these two as two classes which falls under the category of binary classification. In this, one was independent variable (E-mail message column) and the other was a dependent column (receipt).

	email_subject_line	isReceipt
0	Inbox It's on: shop 20-60% off during the Labo...	False
1	Inbox Nike, Lowe's, Best Buy, Macy's Labor Day...	False
2	Inbox We're ready when you are! - Your order i...	True
3	Inbox Macy's, Backcountry, Dyson, HBO Max & Mo...	True
4	Inbox Super-low prices for Labor Day are here ...	False

Figure 3: Sample Input

A2. Preprocessing: Data cleaning is a crucial step in any machine learning model. Without the cleaning process, the dataset is just a cluster of words that the computer does not understand. The preprocessing was done on the loaded data by the removal of special characters, stop words, perform tokenization, stemming and lemmatization.

	email_subject_line	clean_text
0	Inbox It's on: shop 20-60% off uring the Labor...	Inbox It s on shop off uring the Labor Day S...
1	Inbox Nike, Lowe's, Best Buy, Macy's Labor Day...	Inbox Nike Lowe s Best Buy Macy s Labor Day Co...
2	Inbox We're reay when you are! - Your orer is ...	Inbox We re reay when you are Your orer is not...
3	Inbox Macy's, Backcountry, Dyson, HBO Max & Mo...	Inbox Macy s Backcountry Dyson HBO Max & More ...
4	Inbox Super-low prices for Labor Day are here ...	Inbox Super low prices for Labor Day are here ...

Figure 4: Sample Clean Text after preprocessing

Figure 4 shows the original text in column email_subject_line column and clean_text column shows the text after preprocessing is performed on the text data which is email_subject_line column.

A3. Feature Extraction: The features have been extracted using Bag of Words methods. Here the unique vocabulary was created along with their frequency i.e. the number of times a particular word appears in the database.

A4. Vectorization: The raw data, a sequence of symbols (i.e. strings) cannot be fed directly to the machine learning algorithms as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. The most common ways to extract numerical features from text content, namely:

- tokenizing strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.
- Counting the occurrences of tokens in each document.

```
{'inbox': 1602,
 'shop': 2892,
 'labor': 1770,
 'day': 835,
 'sale': 2747,
 'bronze': 393,
 'member': 2030,
 'point': 2430,
 'see': 2818,
 'image': 1592,
 'email': 1034,
 'open': 2252,
 'woman': 3612,
 'men': 2034,
 'home': 1530,
 'handbag': 1450,
```

Figure 5: Words and its frequency

Figure 5 shows the unique words present in the vocabulary and the number of times a particular word occur in that vocabulary.

A5. Classification: After the feature extraction and the raw text were converted into a numerical value, the model were trained using the training dataset and then test the performance of the model with the test dataset. The dataset was split into 70% for training and 30% for testing. Three binary classifiers were used for this experiment and compared. The four-performance metrics were used to evaluate the model performance namely accuracy, precision, recall and F1-score.

VI. RESULTS

A. Performance Metrics

To evaluate the perfomance of the classification model, four performance metrics were used namely Accuracy, Precision, Recall and F1 score.

A1. Accuracy: Accuracy, calculated using (1), represents the number of correctly classified data instances over the total number of data instances.

$$Accuracy = \frac{TN+TP}{TP+TN+FP+FN} \quad (1)$$

Accuracy may not be a good measure if the dataset is not balanced (both negative and positive classes have different number of data instances)

A2. Precision: Precision, calculated using (2) and is defined as:

$$\text{Precision } \frac{TP}{TP+FP} \quad (2)$$

Ideal value of Precision should be 1(high) for a good classifier. It becomes 1 when TP = TP + FP. This means FP is zero. As FP increases the value of the denominator becomes greater than the numerator and the value of the precision decreases.

A3. Recall: Recall, calculated using (3) and is defined as:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Ideal value of Recall should be 1(high) for a good classifier. It becomes 1 when TP = TP + FN. This means FN is zero. As FP increases the value of the denominator becomes greater than the numerator and the value of the recall decreases.

A good classifier is one which has both precision and recall value as one which means that FP and FN should be zero. A metric that takes care of it is F1 Score.

A4. F1 Score:

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

F1 score becomes 1 when both precision and recall equals to 1. It becomes high when both are high. F1 score is the harmonic mean of precision and recall and is a better measure than accuracy.

B. Comparison of Naïve Bayes Bernoulli NB, Support Vector Machines and Random Forest metrics

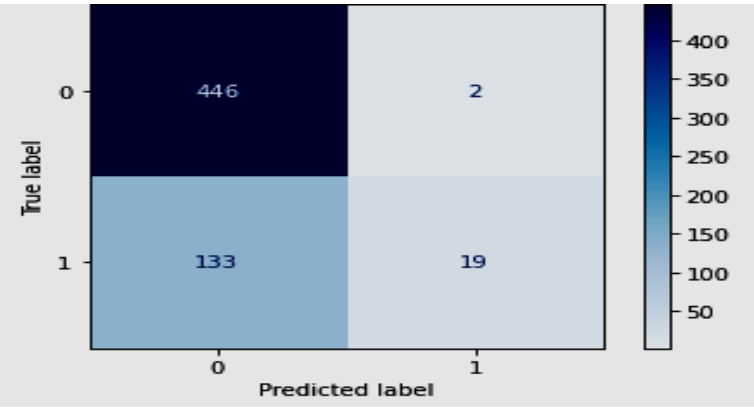
TABLE III COMPARISON OF NAÏVE BAYES BERNOULLI NB, SUPPORT VECTOR MACHINES AND RANDOM FOREST METRICS

Classifier	Accuracy	Precision	Recall	F1
Naïve Bayes Bernoulli NB Random state = 0	73.5%	94.11%	9.19%	16.75%
Random state = 10	76.5	90%	11.46%	20.33%
Random state = 42	77.5%	90.47%	12.5%	21.96
Support Vector Machines Random State = 0	75.33%	100%	14.94%	26.00%
Random State = 10	77.33%	100%	13.37%	23.59%
Random State = 42	78.5%	100%	15.13%	26.28%
Random Forest Random State = 0	80.16%	100%	31.60%	48.03%
Random State = 10	80.5%	100%	25.47%	40.60%
Random State = 42	82.83%	100%	32.23%	48.75%

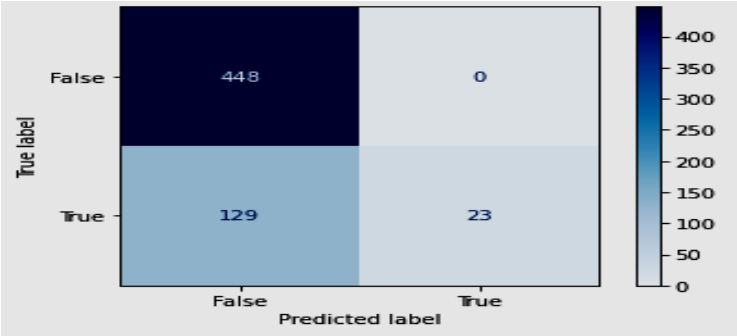
Table III shows the results obtained for the three classifiers with different random state values using the Bag of Words (BOW). It was observed that with random forest having a random state of 42 gives the best results.

C. Confusion Matrix

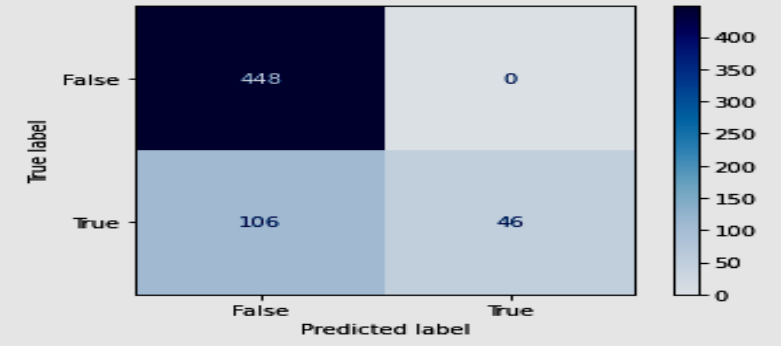
Data is classified into two categories in binary classification: positives (P) and negatives (N). After that, the binary classifier categorizes all data instances as either positive or negative. This classification yields four outcomes: two forms of accurate (or true) classification, true positives (TP) and true negatives (TN), and two types of wrong (or false) classification, false positives (FP) and false negatives (FN) as shown in figure (3,4,5).



-Figure 6: Naïve Bayes Bernoulli NB



-Figure 7: Support Vector Machine



-Figure 8: Random Forest

C1. True Positive (TP): A message which is identified as a receipt (positive) and classified receipt (positive). This is called True Positive (TP). From figure (3, 4, 5), TP values for Naïve Bayes Bernoulli NB, SVM and RF are 19, 23 and 46 respectively.

C2. True Negative (TN): A message which is identified as not a receipt (negative) and classified not a receipt (negative). This is called True Negative (TN). From figure (3, 4, 5), TN values for Naïve Bayes Bernoulli NB, SVM and RF are 446, 448 and 448 respectively.

C3. False Positive (FP): A message which is identified as not a receipt (negative) and classified receipt (positive). This is called **True Negative (*TN*).** From figure (3, 4, 5), FP values for Naïve Bayes Bernoulli NB, SVM and RF are 2, 0 and 0 respectively.

C4. False Negative (FN): A message which is identified as a receipt (positive) and classified not a receipt (negative). This is called **False Negative (*FN*).** From figure (3, 4, 5), FN values for Naïve Bayes Bernoulli NB, SVM and RF are 133, 129 and 106 respectively.

The confusion matrix provides additional information into not just the performance of a predictive model, but also which classes are successfully predicted, which wrongly forecasted, and what types of errors are made.

VII. CONCLUSION AND FUTURE WORK

From Table III, it was observed that among the three binary classifiers, Random Forest with random state of 42 having an accuracy of 82.83% as compared to Support Vector Machine of 78.5% and Naïve Bayes Bernoulli NB of 77.5% found to be more efficient in terms of Accuracy, Precision, Recall and F1-Score. It is also observed that the False Negative value in Random Forest is reduced which is 106 as compared to SVM which is 129 and Naïve Bayes Bernoulli NB which is133. The idea is to have increase in True Positive and decrease in True Negative, False Positive and False Negative also. The future work could be to find more efficient machine learning algorithm which will reduce False Negative values and increase the efficiency of the model.

REFERENCES

[1] Aggarwal, C.C.; Zhai, C. A survey of text classification algorithms. In *Mining Text Data*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 163–222.

[2] Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* 1988, *24*, 513–523.

[3] Goldberg, Y.; Levy, O. Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv* 2014, arXiv:1402.3722.

[4] Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.

[5] Larson, R.R. Introduction to information retrieval. *J. Am. Soc. Inf. Sci. Technol.* 2010, *61*, 852–853.

[6] Manevitz, L.M.; Yousef, M. One-class SVMs for document classification. *J. Mach. Learn. Res.* 2001, *2*, 139–154.

[7] Han, E.H.S.; Karypis, G. Centroid-based document classification: Analysis and experimental results. In *European Conference on Principles of Data Mining and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 424–431.

[8] Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. *JCP* 2012, *7*, 2913–2920.

[9] Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* 2005, *17*, 299–310.

[10] Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* 2005, *17*, 299–310.

[11] Lock, G. Acute mesenteric ischemia: Classification, evaluation and therapy. *Acta Gastro-Enterol. Belg.* 2002, *65*, 220–225.

[12] Pencina, M.J.; D'Agostino, R.B.; Vasan, R.S. Evaluating the added predictive ability of a new marker: From area under the ROC curve to reclassification and beyond. *Stat. Med.* 2008, *27*, 157–172.

[13] Gupta, G.; Malhotra, S. Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example). *Int. J. Comput. Appl.* 2015, *975*, 8887.

[14] Verma, T.; Renu, R.; Gaur, D. Tokenization and filtering process in RapidMiner. *Int. J. Appl. Inf. Syst.* 2014, *7*, 16–18.

[15] Saif, H.; Fernández, M.; He, Y.; Alani, H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.

[16] Spirovski, K.; Stevanoska, E.; Kulakov, A.; Popeska, Z.; Velinov, G. Comparison of different model's performances in task of document classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; p. 10

[17] Sampson, G. *The'Language Instinct'Debate: Revised Edition*; A&C Black: London, UK, 2005.

[18] Plisson, J.; Lavrac, N.; Mladenić, D. A rule based approach to word lemmatization. In Proceedings of the 7th International MultiConference Information Society IS 2004, Ljubljana, Slovenia, 13–14 October 2004.

[19] Korenius, T.; Laurikkala, J.; Järvelin, K.; Juhola, M. Stemming and lemmatization in the clustering of finnish text documents. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004; pp. 625–633

[20] Farzi, R.; Bolandi, V. Estimation of organic facies using ensemble methods in comparison with conventional intelligent approaches: A case study of the South Pars Gas Field, Persian Gulf, Iran. *Model. Earth Syst. Environ.* 2016, *2*, 105.

[21] Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 1999, *36*, 105–139.

[22] Qu, Z.; Song, X.; Zheng, S.; Wang, X.; Song, X.; Li, Z. Improved Bayes Method Based on TF-IDF Feature and Grade Factor Feature for Chinese Information Classification. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018; pp. 677–680.

[23] Soheily-Khah, S.; Marteau, P.F.; Béchet, N. Intrusion detection in network systems through hybrid supervised and unsupervised mining process-a detailed case study on the ISCX benchmark data set. *HAL* 2017.

[24] Wang, Y.; Khardon, R.; Protopapas, P. Nonparametric bayesian estimation of periodic light curves. *Astrophys. J.* 2012, *756*, 67.

[25] Bo, G.; Xianwu, H. SVM Multi-Class Classification. *J. Data Acquis. Process.* 2006, *3*, 017.

[26] Manevitz, L.M.; Yousef, M. One-class SVMs for document classification. *J. Mach. Learn. Res.* 2001, *2*, 139–154.

[27] Bansal, H.; Shrivastava, G.; Nhu, N.; Stanciu, L. *Social Network Analytics for Contemporary Business Organizations*; IGI Global: Hershey, PA, USA, 2018