

Classifying an Incoming Customer Message into Spam Versus Ham

Vivek Astvansh*

Associate Professor of Quantitative Marketing and Analytics,
Desautels Faculty of Management, McGill University, Canada
vivek.astvansh@mcgill.ca

Academic Director, Bensadoun School of Retail Management, McGill University, Canada

Adjunct Associate Professor of Data Science, Department of Informatics,
Luddy School of Informatics, Computing, and Engineering,
Indiana University Bloomington, USA |

Affiliate, Environmental Resilience Institute,
Indiana University

**Acknowledgments: Vivek wrote this manuscript while he was the Research Director of the Indiana University Bloomington Kelley School of Business' Center for Education and Research in Retail. He thanks the Center's Director, John Talbott, for (1) introducing him to the retailer who provided the data and (2) paying Gautam Chauhan (a master's student at IU's Luddy School), who assisted Vivek in the research.*

Classifying an Incoming Customer Message into Spam Versus Ham

ABSTRACT

Customers often communicate with a company by sending it emails, filling out the “contact us” form on its website, and posting messages on social media platforms. Unfortunately, a substantial part of these incoming customer messages can be spam. Whether an email is authentic (i.e., “ham”) or spam is specific to the company. Consequently, commercial spam classifiers (e.g., the one Microsoft Outlook uses) do not work well. A machine learning-based spam classifier can solve this problem. The author used a large data set from a publicly traded retailer in the United States to test 27 TF-IDF-based and six word-embedding-based binary classifiers. He found that RoBERTa—a sophisticated embedding-based classifier—provided the lowest false positive rate of 5.31%. However, its rival classifier—XLNet—offered marginally superior (92%) balanced accuracy, relative to RoBERTa’s 91%. To enhance the use of the models by other organizations and academics, the author offers supplemental models trained on only an external data set and a combination of internal and external data sets. The manuscript contributes by applying a broad set of existing machine-learning models to solve a real problem for a real company. The author publishes the code (via the journal), which other companies and researchers can use.

Keywords: Spam, machine learning, classifier, KeyBERT, LDA

INTRODUCTION

Customers often seek support from a company by sending it emails, filling out “contact us” forms on the company’s app or website, and texting the company on social media platforms and messaging apps. The company’s employees—in the customer service department—read and respond to these messages.¹ However, an increasing proportion of incoming customer messages is spam (Johnson 2021), which causes two problems for the company. First, the mere opening of a spam email confirms to the sender that the receiving email address is active. The mere opening could install malware on the employee’s computer and potentially the company’s network. For example, Facebook and Google are reported to have lost \$121 million when an email sender impersonated a vendor.² Second, manually classifying each message into spam versus ham (i.e., an authentic message, or a “nospam”) can frustrate employees (Smith 2020) and waste their time, raising the company’s costs of serving customers. Third, manual classification is subjective and may lead to false positives³ (Southeastern Technical 2020), leaving some customer messages unanswered. Statista estimates the total cost of spam to reach US\$16 trillion by 2029. This problem elicits the following managerially consequential question: *How can a company separate ham from spam?*

I partnered with a U.S.-based, NASDAQ-listed retail company (that prefers to stay anonymous) to automate classifying its incoming customer messages into spam versus ham. I met with the VP of Marketing and three of their direct reports (title: Director) who managed customer service, technology, and operations. I also met with the VP of consumer technology

¹ Commercially available spam classifiers—such as the one that Microsoft Outlook uses—classify emails into spam vs. nonspam (i.e., ham) by the rarity of the “from” email address and not by their content. Because most of the messages a company receives are from “new” addresses, these commercial classifiers predict most messages as spam, leading to high false positive rate. Consequently, companies prefer to develop their own classifier, which is specific to the problem of incoming customer messages (as opposed to emails received by individuals) and preferably, learn from the hidden context on why customers message the company.

² <https://www.teramind.co/blog/business-email-compromise-examples/>

³ A false positive is a ham customer message that a representative may incorrectly classify as spam. The classification is subjective, unless the company writes rules for classification, trains its representatives who classify incoming messages, and checks the quality of such classification.

and innovation to understand their future-oriented perspective. Lastly, the company matched me with a manager of their customer service function, who became my point of contact. These conversations help me understand their problems, the data they have, and how I can apply data science to solve their problem. I received from the customer service manager the census of 148,104 customer messages it received (as emails and filled “contact us” form on the website/app) (I call the company-proprietary data set “internal”). Each message was assigned to a service agent. The representatives had labeled 5,826 of these messages as spam. The customer service manager informed me that the company prefers to avoid false positives while monitoring false negatives, true positives, and true negatives. I thus chose the *false positive rate* and *balanced accuracy* to compare the performance of my machine learning classifiers (Boghrati and Berger 2021).

I proceeded in three steps, running three types of classifiers, all trained and tested on the internal data set: (1) 27 TF-IDF-based (and thus the foundational) classifiers, (2) a rudimentary word embedding method-based logistic regression, random forest, and support vector machine (SVM) classifiers, (3) the more sophisticated word embedding-based XLNet and RoBERTa (Puranam, Kadiyali, and Narayanan 2021).

First, following foundational and recent research in machine learning, I began by training the 27 most common classifiers, using *term frequency*, *inverse document frequency* (TF-IDF)⁴ on the internal data set. Among these 27 classifiers, the support vector machine (SVM) classifier offers the highest balanced accuracy of 87.25% on the “test” data set. Perhaps unsurprisingly, the

⁴ TF-IDF is a numerical measure of the importance of a term (e.g., a word) in a document, given a corpus of multiple documents. Simply put, the importance of a term in a document, given a corpus, is proportional to its frequency in the focal document (i.e., term frequency) but inversely proportional to the number of documents in which it appears (i.e., inverse document frequency). Stated differently, this method rewards a term that occurs frequently in a document but penalizes it if it appears in multiple documents.

SVM classifier ranks the most accurately on three other performance measures: accuracy, the area under the receiver operator characteristic (ROC) curve, and the F1 score.

Second, the above SVM (and the other 26 classifiers) suffer from two limitations: (1) the use of the TF-IDF, which weighs a term the same regardless of its surrounding terms (i.e., the context), and (2) not tune the hyperparameters used to train the classifier. I overcame both limitations by using a *word-embedding* method. Specifically, I overcame the first limitation using Google’s Bidirectional Encoder Representations from Transformers (BERT) instead of TF-IDF (Devlin, Chang, and Toutanova 2018), trained on the internal data set. Unlike TF-IDF, a BERT considers the context in which a term is used and thus assigns different weights to a term based on its neighboring terms.⁵ I overcome the second limitation using a grid search, which uses 80% of observations in my “sample” data set for training and the remaining 20% for validation. The validation step tunes the hyperparameters before producing the trained model. Next, I use a separate “test” data set for prediction. I developed a BERT-based SVM classifier because the SVM classifier was the most accurate among the 27 TF-IDF classifiers. This SVM classifier produced a false positive rate of 13.49% and a balanced accuracy of 87%. Interestingly, the BERT-based classifier (with 87% balanced accuracy) is no more accurate than the TF-IDF-based classifier (which produced an 87.25% balanced accuracy). A logistic regression and a random forest are popular alternatives to an SVM. Therefore, I also trained—again, using company-proprietary or the internal data set—a BERT-based logistic regression and a random forest. I compared the performance of these two classifiers against the BERT-based SVM. The random forest produces the lowest false positive rate of 7.14%, whereas the SVM is the most

⁵ BERT uses “masked-language modeling” (MLM) and “next sentence prediction” (NSP). Simply put, BERT takes as input a sentence that has a few tokens/terms masked and next attempts to produce the same sentence as output. This method is similar to human beings performing a “fill in the blank,” such as “In the fall, the ____ fall from the trees.” This method allows BERT to understand how the language is used for communication.

accurate in balanced accuracy (87%). Next, I integrated the outputs from the three BERT-based classifiers by running a voting classifier, which produced a false positive rate of 11.72% and a balanced accuracy of 85%.

Third, marketing research has considered three transformer models: BERT (Devlin, Chang, Lee, and Toutanova 2018), XLNet (Yang et al. 2019), and RoBERTa (Liu et al. 2019) (Nguyen, Johnson, and Tsiros 2024; Puranam, Kadiyali, and Narayan 2021). Further, XLNet and RoBERTa⁶ have been shown to outperform BERT (Puranam, Kadiyali, and Narayan 2021). Therefore, I trained these two models next. XLNet offered a balanced accuracy of 92% and a false positive rate of 5.98%. The corresponding numbers for RoBERTa are 91% and 5.31%. Therefore, XLNet and RoBERTa outperform the BERT-based classifiers on all accuracy rates. In summary, should the false positive rate be the exclusive criterion, my partner company should use the RoBERTa (5.31% false positive rate). However, if the company cares about balancing all four accuracy rates (false positive, false negative, true positive, and true negative), the XLNet is marginally superior (92%) to RoBERTa (91%).

A downside of my models above is that they all are trained on my partner company's (i.e., the "internal") data set. While such a training data set allows the models to learn context-specific rules, the generalizability of the trained model on data sets from other data-generating processes can be low. I overcame this limitation by running two types of supplementary models. Specifically, in the first type of supplementary model, I repeated step 2 (BERT-based four classifiers: logistic regression, random forest, SVM, and voting classifier) and step 3 (XLNet and RoBERTa) articulated above, training the models on a combination of two *external* (i.e., publicly

⁶ RoBERTa expands to robustly optimized BERT pretraining approach.

available, or open source) data sets and testing them on the company's data set. XLNet offered a 1% false positive rate, while XLNet and RoBERTa offered 98% balanced accuracy.

In the second type of supplementary model, I repeated steps 2 and 3, training the models on a combination of the internal *and* the external data sets while testing the model on the company's data set. BERT-based random forest and RoBERTa offered a false positive rate of 4%, while both XLNet and RoBERTa scored 93% on balanced accuracy.

I complement the supervised ML models (the classifiers) by building topic models—an unsupervised suite of models. I aimed to discover topics underlying ham and spam. I built the conventional LDA and KeyBERT + LDA. I could not label the topics produced by these models for the spam data set, supporting the intuition that spam is truly junk. I built these two LDA models and a guided LDA for the ham data set. I could label the topics produced by these models.

The complete exercise helps me draw four insights. (1) Regardless of the accuracy measure one considers, XLNet and RoBERTa outperform BERT classifiers, which outperform TF-IDF-based classifiers. (2) An SVM is more accurate than a logistic regression and a random forest, regardless of whether it is trained using a TF-IDF or a BERT model. (3) A voting classifier is only sometimes—and not consistently—more accurate than each constituent classifier. (4) Spam is junk, with no topics that apply to the corpus.

The findings inform managers of a pecking order of the various ML models. Specifically, I show that XLNet and RoBERTa perform better than the BERT-based classifiers, which outperform TF-IDF-based classifiers. This order will help managers save costs in building and testing various models and instead focus on the models that are likely to produce the best performance. I have presented the findings to my partner company and provided them with my

software program (Python file). The company, in turn, has asked its information technology (IT) integration vendor to apply the program in the production environment.⁷ More importantly, following recent research (Hovy, Melumad, and Inman 2021; Jededi et al. 2021; Warren et al. 2021), I publish my machine learning models (the Python source code file)—trained on the external data set and a combined internal + external data set—that managers from other organizations⁸ and researchers can download from the journal’s website and use it to automate their spam classification. Indeed, I have suggested that my partner company’s customer service partner apply my Python program in its test environment and measure the potential savings in operational costs (Fang, Gao, and Hu 2021) and service employees’ satisfaction. Ideally, I would have preferred to provide these validations of my model, but such validations could have taken months.

Substantively, my research contributes to the modeling literature on customer service. This literature not only uses optimization models and game theory (e.g., Delana, Savva, and Tezcan 2021; Sun, Argon, and Ziya 2022) but also models the business phenomena using empirical data (Altman et al. 2021; Hu, Allon, and Bassamboo 2022). I contribute to the literature by *applying* existing machine learning methods to solve a real business problem for a real company and publishing my code for broader use and potential extensions.

Methodologically, my research belongs to the emerging literature that applies machine learning models to understand and predict business phenomena and outcomes (Berger et al. 2020; Berger et al. 2022; Hartmann and Netzer 2023; Lee, Chakraborty, and Banerjee 2023; Liu 2023; Padmanabhan et al. 2022; Puranam, Kadiyali, and Narayan 2021). Lastly, in applying and

⁷ My partner company recently acquired another retailer. Consequently, the vendor is focused on integrating the acquired company on my partner’s IT infrastructure and therefore, the implementation of my classifier is delayed.

⁸ The vendor’s exclusive business is to contract customer service for retailers in the United States. The vendor’s representative has told me that after (and if) he has evidence that my program improves its performance, it will apply the program in its production environment of other clients.

comparing a broad set of binary classifiers on internal, external, and combined data sets, I also add to the extant machine learning literature on classifiers (Bhopale and Tiwari 2021; Blamzieri and Bryl 2008; Dada et al. 2019; Nisar, Rakesh, and Chhabra 2021; Srinivasan et al. 2021; Yaseen 2021).

Next, I present the context of my partner company, the data, and the methods. I follow up with a listing of the accuracy measures. While all report all tables and figures for the classifiers, I relegate the details to the web appendices.

CONTEXT, DATA, AND METHOD

My partner company is a U.S.-headquartered NASDAQ-listed footwear retailer. It operates nearly 400 stores in the Midwest, south, and southeast regions. The company was founded in 1978 and earned a record sales revenue of \$1.175 billion in the fiscal year 2023. In 2023, 49% of the firm's net unit sales were in the nonathletic categories, 45% in athletics, 5% in accessories, and 1% in others. Further, Nike, Skechers, and Crocs collectively accounted for 45% of the firm's net sales.

The company has outsourced its customer service to a Minneapolis-based company specializing in providing customer service to U.S.-based retailers. My partner company uses Salesforce Service Cloud software to manage customer service data. Whenever the company receives a message on any communication channel, the Salesforce Service Cloud program automatically creates a "case."

I received from the company the population of 148,104 unique customer service *cases* that the company received. These cases originated as either email messages or filled "contact us" forms on the company's website. The Salesforce Service Cloud identifies each case uniquely by a *Case Number*. Each case has a customer-provided textual *Description*. The customer service

representative (whom the retailer calls a *Case Owner*) reads the *Description* to determine whether the case is spam. If the representative believes the message is spam, they mark it as spam in the *Case Comments* column. Of the 148,104 cases in my data set, 5,826 (3.93%) were spam.

Training the Classifiers

I aim to identify and create the most accurate machine learning classifier that automatically classifies customer messages into spam versus ham. Creating such a classifier requires training the classifier first. Machine learning models can be trained “externally” or “internally.” External training involves training the classifier on an external data set sourced from a context that likely has a different data-generating process. In contrast, internal training means training the classifier on randomly chosen observations from my partner company’s data set. Each type of classification has its benefits and costs. Whereas an internally trained classifier can learn context-specific rules, an externally trained classifier would be agnostic to the context and thus more generalizable. Which of these two types of training produces a more accurate classifier is an empirical question that I attempt to answer by training my classifiers on *only an internal data set*, *only an external data set*, and a combination of the *internal and external data sets*.

Training the Classifiers on Only the Internal Data Set

Because I aim to compare my classifiers trained on an internal data set with those trained on an external data set, the number of observations I use for training must be comparable. Each of my two external data sets has about 4,000 *distinct* emails. Therefore, of the 148,104 observations provided by the company, I used 4,000 as my “sample” data set, which a classifier uses to train itself and subsequently validate itself by tuning its hyperparameters. My next task

was identifying how many of these 4,000 observations would be spam and how many ham. My two external data sets classify 30% of observations as spam and 70% as ham. I used this distribution to include in my internal data set 1,200 messages that were classified (by values of *CaseComments* and *Description*) as spam and 2,800 that were ham.

The remaining 144,104 (148,104 – 4,000) observations serve as my “test” data set, on which I run the classifier and predict whether an observation is spam or ham.

Training the Classifiers on Only the External Data Set

I searched the following six databases for external (publicly available) data sets: Google Dataset Search (<https://datasetsearch.research.google.com/>), Kaggle (<https://www.kaggle.com/datasets>), Data.Gov (<https://www.data.gov/>), Datahub.io (<https://datahub.io/>), UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/index.php>), and Zenodo (<https://zenodo.org/>). I found two data sets that serve my purpose. First is the Spam Filter data set (<https://www.kaggle.com/karthickveerakumar/spam-filter?select=emails.csv>), which lists the text of 5,695 unique emails, of which 1,368 (23.88%) are spam. The second is the Spam Mails data set (<https://www.kaggle.com/venky73/spam-mails-dataset>), which includes text for 4,993 distinct emails, of which 1,499 (28.99%) is spam. I merged the two data sets and removed duplicates to realize a combined data set of 10,140 observations, which I used as my external data set to train the ML classifiers.

I used 148,104 observations in my internal data set for prediction. These observations serve as my “test” data set when I use external data sets to train my classifier.

Training the Classifiers on the Combined (Internal + External) Data Set

Lastly, I combined the internal data set and its external counterpart to create a “combined” data set. I trained all my BERT-based classifiers—logistic regression, random forest, SVM, voting classifier, XLNet, and RoBERTa—on the combined data set.

Which Measure of Accuracy Does My Partner Company Prefer?

I interviewed the customer service manager to understand whether the company considers all four accuracy measures— true positive, true negative, false positive, and false negative—equally important or prioritizes them in some order. The manager told me the company prefers to avoid false positives because they indicate that the customer service representative should ignore the customer’s message. If the customer does not receive a response from the company, they may become dissatisfied and choose not to buy from it again. Next, the company weakly prefers to avoid false negatives because they would waste a representative’s time. The company is indifferent to achieving high accuracy rates for true negative and true positive.

ML researchers suggest that in classifications where all four first-order accuracy measures are equally important, one can use balanced accuracy or an F1 score. However, if one of the four rates is more critical (e.g., false positive in my context), balanced accuracy is more appropriate than the F1 score (<https://neptune.ai/blog/balanced-accuracy>). Consequently, I consider balanced accuracy to be my measure of accuracy. However, because the false positive rate is the most important among the four first-order accuracy measures, I also consider this rate to compare my classifiers.

Preprocessing

The data set I received from the company was already preprocessed. Specifically, it did not include punctuation marks, numbers, or special characters. However, I performed three specific preprocessing tasks. First, I removed one-character long words. Second, I dropped stop words listed in the NLTK library. Third, I lemmatized the remaining words. After these three preprocessing steps, I created a document-term matrix using the TF-IDF.

ACCURACY RATES FROM THE 27 TF-IDF-BASED CLASSIFIERS (TRAINED ON ONLY THE INTERNAL DATA SET)

I used 27 machine learning models used commonly to classify binary outcomes. Further, I relied on my internal data set to train each classifier. Specifically, I passed my sample data set of 4,000 observations to each classifier. I configured the classifier to randomly select 70% of these observations (i.e., 2,800) for training itself and the remaining 30% of observations (i.e., 1,200) for testing/prediction. Table 1 reports the four second-order accuracy measures for the 27 classifiers. Because balanced accuracy is the most suitable measure for my data set, I sorted the table by descending values of balanced accuracy. The result suggests that a support vector machine (SVM) is the most accurate classifier based on balanced accuracy (value 87.25%), accuracy, AUC ROC, and F1 score.

Table 1: Four Accuracy Rates for 27 TF-IDF-Based Classifiers

Note: The rows are sorted by descending values of Balanced Accuracy. Source: Author's creation

Model	Accuracy	Balanced Accuracy	(Unweighted) F1 Score	AUC
SVM	0.8813	0.8725	0.8827	0.8725
Nu SVM	0.8750	0.8551	0.8754	0.8551
LGBM Classifier	0.8613	0.8323	0.8608	0.8323
Ridge Classifier CV	0.8550	0.8313	0.8555	0.8313
Linear Discriminant Analysis	0.8500	0.8289	0.8510	0.8289
Ridge Classifier	0.8513	0.8275	0.8518	0.8275
Nearest Centroid	0.7963	0.8271	0.8042	0.8271
XGB Classifier	0.8513	0.8204	0.8508	0.8204

Bernoulli NB	0.7838	0.8169	0.7924	0.8169
Gaussian NB	0.7925	0.8138	0.8002	0.8138
Logistic Regression	0.8350	0.8029	0.8348	0.8029
Perceptron	0.8188	0.7971	0.8208	0.7971
Passive Aggressive Classifier	0.8325	0.7928	0.8310	0.7928
Kneighbors Classifier	0.8313	0.7884	0.8292	0.7884
Random Forest Classifier	0.8325	0.7810	0.8286	0.7810
Linear SVC	0.8075	0.7808	0.8092	0.7808
Ada Boost Classifier	0.8138	0.7794	0.8139	0.7794
SGD Classifier	0.8238	0.7783	0.8214	0.7783
Extra Trees Classifier	0.8250	0.7709	0.8207	0.7709
Calibrated Classifier CV	0.8213	0.7482	0.8119	0.7482
Bagging Classifier	0.8075	0.7466	0.8018	0.7466
Decision Tree Classifier	0.7525	0.7108	0.7536	0.7108
Extra Tree Classifier	0.7313	0.6849	0.7323	0.6849
Quadratic Discriminant Analysis	0.7013	0.5171	0.6032	0.5171
Dummy Classifier	0.5875	0.5089	0.5870	0.5089
Label Spreading	0.6988	0.5035	0.5816	0.5035
Label Propagation	0.6988	0.5035	0.5816	0.5035

The above SVM classifier—and each of the other 26 classifiers—is limited in two ways. First, it trains itself using a TF-IDF matrix, which assumes that a word has a fixed meaning, irrespective of the context of the sentence in which it appears. Second, it does not fine-tune its hyperparameters. I overcome the first limitation by using Google’s Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, and Toutanova 2018) to train my SVM classifier. In simpler words, BERT reads the entire sentence (i.e., the context) and assigns an embedding to a word. I tune the hyperparameters to overcome the second limitation.

ACCURACY RATES FROM FOUR BERT-BASED CLASSIFIERS TRAINED ON ONLY THE INTERNAL DATA SET

I used the grid search to tune the model’s hyperparameters. Internally, grid search splits the *sample* data set of 4,000 observations into an 80-20 ratio, such that 80% (i.e., 3,200) of randomly selected observations serve as my *training* data set (on which it trains the classifier).

The remaining 20% of observations serve as the *validation* data set (on which it applies the trained classifier and determines the optimal values of the parameters).

Next, I applied this trained classifier to the 144,104 observations in my test data set. I obtained the values of the four first-order accuracy rates: false negative, false positive, true negative, and true positive. I used these four rates to obtain values of the four second-order accuracy rates.

Because logistic regression and random forest are popular alternatives to the SVM classifier⁹, I first run these two classifiers before focusing on the SVM classifier. Web Appendix A provides each classifier's confusion matrix, classification report, and ROC curve. Because my partner company prefers to avoid false positives, I first compare the three classifiers on the false positive rate. The logistic regression's false positive rate is 15.20%, whereas the random forest's rate is 7.14%. Interestingly, the SVM classifier produces a false positive rate of 13.49%. An exclusive focus on the false positive rate suggests that random forest trumps logistic regression and SVM classifier. However, I focus on the balanced accuracy measure because my partner company prefers to avoid false negatives and values true negatives and true positives (albeit being indifferent to their relative importance of the two metrics). I note that the logistic regression produces a balanced accuracy of .86. In contrast, the random forest provides a balanced accuracy of .76. The SVM trumps these classifiers by yielding a balanced accuracy of .87.

I use a voting classifier to combine the results from these three classifiers. As the name suggests, a voting classifier considers the vote from multiple models and classifies a given observation based on the majority vote. For example, if two of my three classifiers label a

⁹ I did not consider naïve Bayes classifier because it requires only nonnegative values in vectors. Because BERT produces vectors with positive and negative values, a naïve Bayes classifier does not work with BERT.

message as spam and one classifier calls it ham, the voting classifier would use the majority votes and tag the message as spam. Tables 2A and 2B below provide the voting classifier's confusion matrix and classification report. Figure 1 provides the ROC curve. Interestingly, the voting classifier offers a false positive rate of 11.72%, lower than the SVM's rate of 13.49% but higher than the random forest's rate of 7.14%. Further, the voting classifier provides a balanced accuracy of .85, slightly lower than the .87 balanced accuracy of the SVM classifier.

Table 2A: Confusion Matrix for the *Voting Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		Ham	Spam
Ham	139,477 (96.78%)	123,137 (True Negatives) 88.28%	16,340 (False Positives) 11.72%
Spam	4,626 (3.22%)	853 (False Negatives) 18.43%	3,773 (True Positives) 81.57%
Total	144,103	123,990	20,113

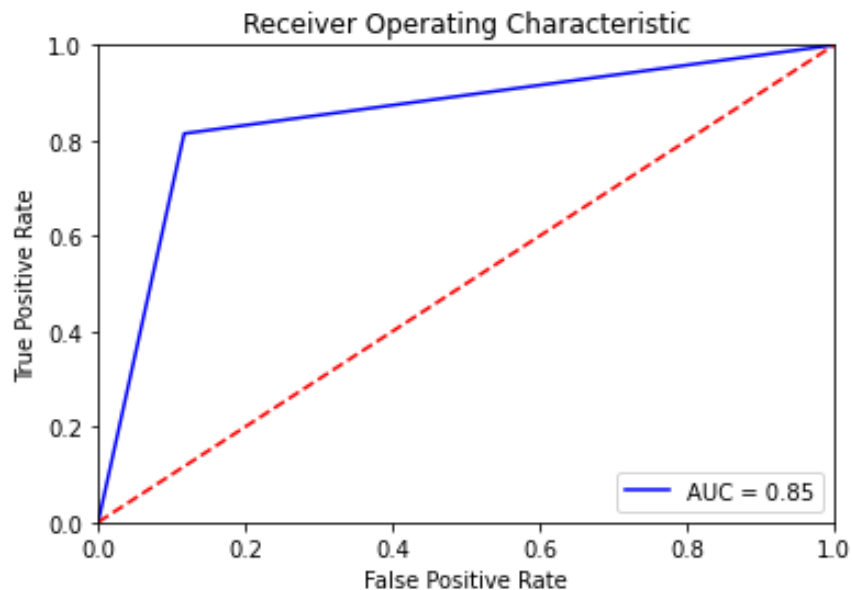
Table 2B: Classification Report for the *Voting Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.88	.93
1 (Spam)	.19	.82	.31
Accuracy	.88		
Balanced Accuracy	.85		

Figure 1: ROC Curve for the *Voting Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation



ACCURACY RATES FROM XLNET AND ROBERTA TRAINED ON ONLY THE INTERNAL DATA SET

Deep learning has led to recent breakthroughs in text classification algorithms. One state-of-the-art algorithm is XLNet, an auto-regressive language model that understands sequential data. For example, consider the sentence, “New York is a city.” XLNet creates a sequence of tokens [New, York, is, a, city] and learns to predict probability $\Pr(\text{New} \mid \text{York, is, a, city})$. That is the probability that “New” will be in the sequence given (York, is, a, city) present in the sequence.

I chose XLNet because it has been shown to outperform Google’s BERT model on 20 natural language processing tasks (Yang et al. 2019). I used a pretrained model of XLNet trained on 3.3 billion English words and fine-tuned it for our purpose of spam versus ham classification. Before loading our internal data set into the pretrained XLNet model, I undersampled my data set, using 20,826 observations after randomly shuffling the data set. My internal data set suffers from the class imbalance problem. Therefore, of the 20,826 observations, 15,000 belonged to the ham class and 5,826 to the spam class. Further, I reserved a hold-out set to validate or test the

accuracy of the trained model. Specifically, I used 80% (i.e., 16,660) of my observations for training the XLNet and the remaining 20% (i.e., 4,166) observations for testing. The model achieved a balanced accuracy of 92% and produced a false positive rate of 5.98%. Tables 3A and 3B report the confusion matrix and classification report of our XLNet. Figure 2 shows the ROC curve.

Table 3A: Confusion Matrix for *XLNet* Trained on *Only the Internal Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the XLNet Classifier using Random Subsampling		
		Ham	Spam
Ham	3,012 (73%)	2,832 (True Negatives) 94.02%	180 (False Positives) 5.98%
Spam	1,154 (27%)	119 (False Negatives) 10.3%	1035 (True Positives) 89.68%
Total	4,166	2,951	1,215

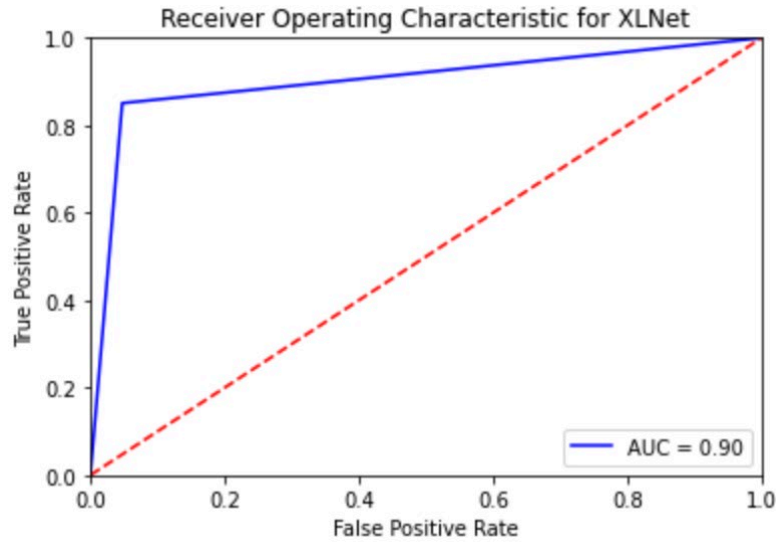
Table 3B: Classification Report for *XLNet* Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.96	.94	.95
1 (Spam)	.85	.90	.87
Accuracy	.93		
Balanced Accuracy	.92		

Figure 2: ROC Curve for the *XLNet* Trained on *Only the Internal Data Set*

Source: Author's creation



Another state-of-the-art model that has proven to offer a performance improvement over Google’s BERT model is RoBERTa. RoBERTa is essentially an extension of the BERT model with a different training approach. Researchers at Facebook found that BERT was significantly undertrained (Liu et al. 2019) and decided to add approximately 160 GB of training data to the model. RoBERTa ended up outperforming BERT by 2% to 20%. Again, I randomly shuffled the internal data set and undersampled it before feeding it into a pretrained RoBERTa model. I chose 20,826 data points comprising 15,000 ham messages and 5,826 spam messages. RoBERTa gave a balanced accuracy of 91% and a false positive rate of 5.31% (see Tables 4A and 4B). Figure 3 illustrates the ROC curve.

Table 4A: Confusion Matrix for RoBERTa Trained on Only the Internal Data Set

Source: Author’s creation

Classified by the Case Owner	Classified by the RoBERTa Classifier using Random Subsampling		
		Ham	Spam
Ham	3,012 (73%)	2,852 (True Negatives) 94.68%	160 (False Positives) 5.31%
Spam	1,154 (27%)	131	1023 (True Positives)

		(False Negatives) 11.35%	88.64%
Total	4,166	2,951	1,215

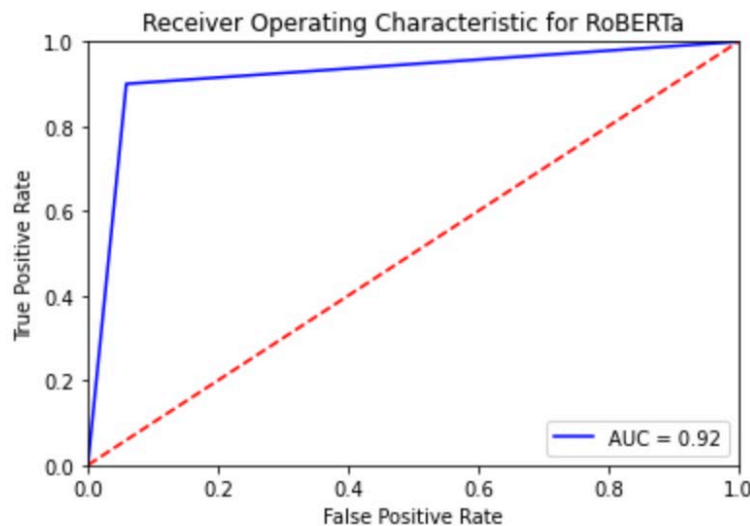
Table 4B: Classification Report for *RoBERTa* Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.95	.94	.95
1 (Spam)	.86	.88	.87
Accuracy	.93		
Balanced Accuracy	.91		

Figure 3: ROC Curve for the *RoBERTa* Trained on *Only the Internal Data Set*

Source: Author's creation



SUPPLEMENTARY MODELS

Accuracy Rates from the Six Classifiers Trained on *Only the External Data Set*

The upside of training the ML classifiers on only the internal data set is that the model learns context-specific rules. The downside, however, is that the model is arguably less

generalizable. One way to increase generalizability is to use one or more external data sets and check whether balanced accuracy increases. To achieve this aim, I used 10,140 observations from <https://www.kaggle.com/karthickveerakumar/spam-filter> and <https://www.kaggle.com/venky73/spam-mails-dataset> as my *sample* data set. Web Appendix B provides the confusion matrix, classification report, and the ROC curve for the six classifiers—logistic regression, random forest, SVM, voting classifier, XLNet, and RoBERTa—trained only on the external data set.

Figure 4: Balanced Accuracy and False Positive Rate for the Six Models Trained on *Only* the *External* Data Set

Source: Author's creation



Figure 4 reports that XLNet produces the lowest (.69%) false positive rate among the six classifiers, whereas both XLNet and RoBERTa provide a balanced accuracy rate of 98%. I am

pleasantly surprised by the remarkably low false positive rate, although I expected a high, balanced accuracy.

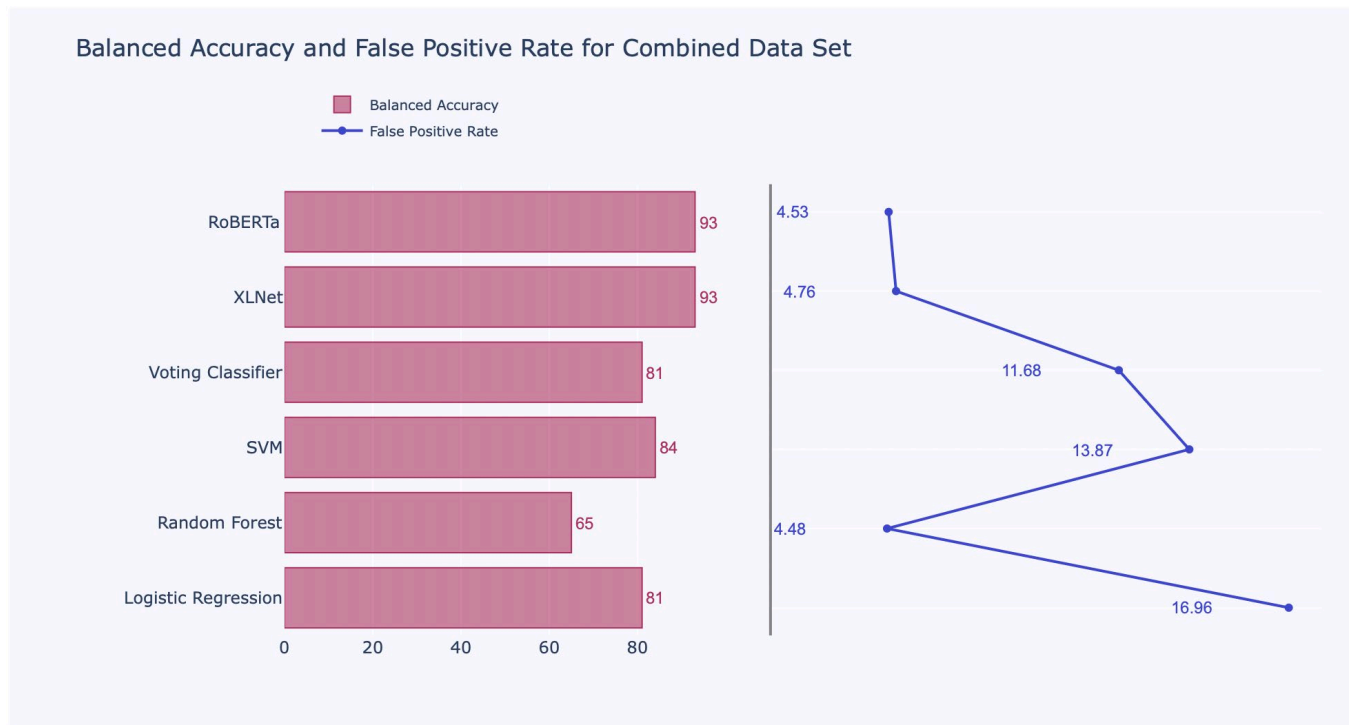
Accuracy Rates from the Six Classifiers Trained on *the Combined Data Set*

One could reason that training a classifier on a combination of internal *and* external data sets can overcome the downside while retaining the upside. I explored this reasoning.

Specifically, I combined the 10,140 observations in my external data set with 4,000 internal observations to create my combined *sample* data set. Next, I trained the three classifiers on the combined data set and predicted whether each of the 144,104 observations in my test data set was spam or ham.

Figure 5: Balanced Accuracy and False Positive Rate for the Six Models Trained on the *Combined Data Set*

Source: Author's creation



Web Appendix C offers the confusion matrix, classification report, and ROC curve for the six classifiers, while Figure 5 depicts the false positive rate and balanced accuracy.

Surprisingly, random forest's false positive rate of 4.48% is marginally lower than that of XLNet (4.76%) and RoBERTa (4.53%). Unsurprisingly though, both XLNet and RoBERTa offer 93% balanced accuracy.

Do the Spam and Ham Vary in Content?

While using a supervised machine learning model—the classifiers—helps, one can receive additional insight by using an unsupervised model, specifically, the topic model. I separated my data set of 148,104 emails into two data sets: a spam data set comprising 5,826 (4%) emails and a ham data set of 142,277 (96%) emails. I performed three sequential preprocessing tasks on each data set. First, I removed all punctuation marks, stop words, and HTML tags. Second, I lemmatized the text. Third, I dropped all words that occurred in fewer than 10 emails or more than .9% of emails. I built two types of topic models for each data set. The first is the conventional LDA, which operates at the word level and uses the entire text of an email. For the second, I ran a BERT-based keyword extractor—KeyBERT—and replaced each email with the topmost unigram or the topmost bigram produced by KeyBERT. I fed the resulting emails to the conventional LDA. Web Appendix D narrates the details and provides the exhibits.

The key insight is that—regardless of the LDA models—I could not make sense of the keywords extracted from the spam data set. Paradoxically, the lack of keywords in spam makes sense because they are, by definition, junk emails. In contrast, the keywords from the ham data set make sense. Consequently, I built a third type of LDA model for the ham data set. Specifically, I took the keywords extracted from KeyBERT and inputted them into a guided LDA model. I received three topics: order, email, and payment.

DISCUSSION

By some estimates, nearly 50% of the contemporary traffic on the internet is spam (Johnson 2021), and this traffic is expected to grow in the future. Unsurprisingly, not only individuals but also companies receive spam messages. One area where these spam messages waste company resources is customer service. Companies, such as my partner company, ask their customer service employees to read the incoming messages manually and classify them into spam versus ham. Two problems characterize this process. First, the classification is subjective. Indeed, when I conversed with the employees of my partner company, I was told that the employees use intuition rather than rules. When I probed further, I heard that although the company dislikes false positives, it is willing to tolerate the errors caused by subjective classification. The second problem is that manual classification frustrates employees and wastes their time. A direct consequence of this method is that employees engage in work that is not what they are hired and paid to do. My research solves these problems for my partner company and other organizations who face the same problem.

Table E1 (in Web Appendix E) lists the eight accuracy rates for each of the six classifiers' three types—trained using only internal, only external, and combined data sets. Figures 6A and 6B show the false positive rate and balanced accuracy for the 18 classifiers.

Figure 6A: False Positive Rate for Different Models Trained on *Only the Internal Data Set*, *Only the External Data Set*, and the *Combined Data Set*

Source: Author's creation

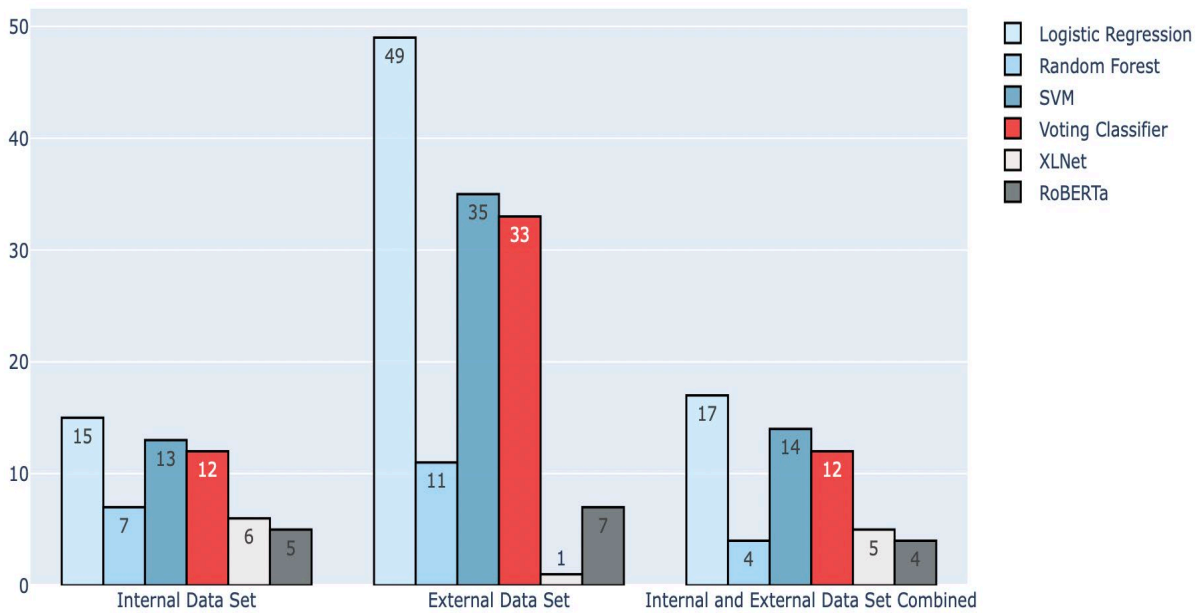
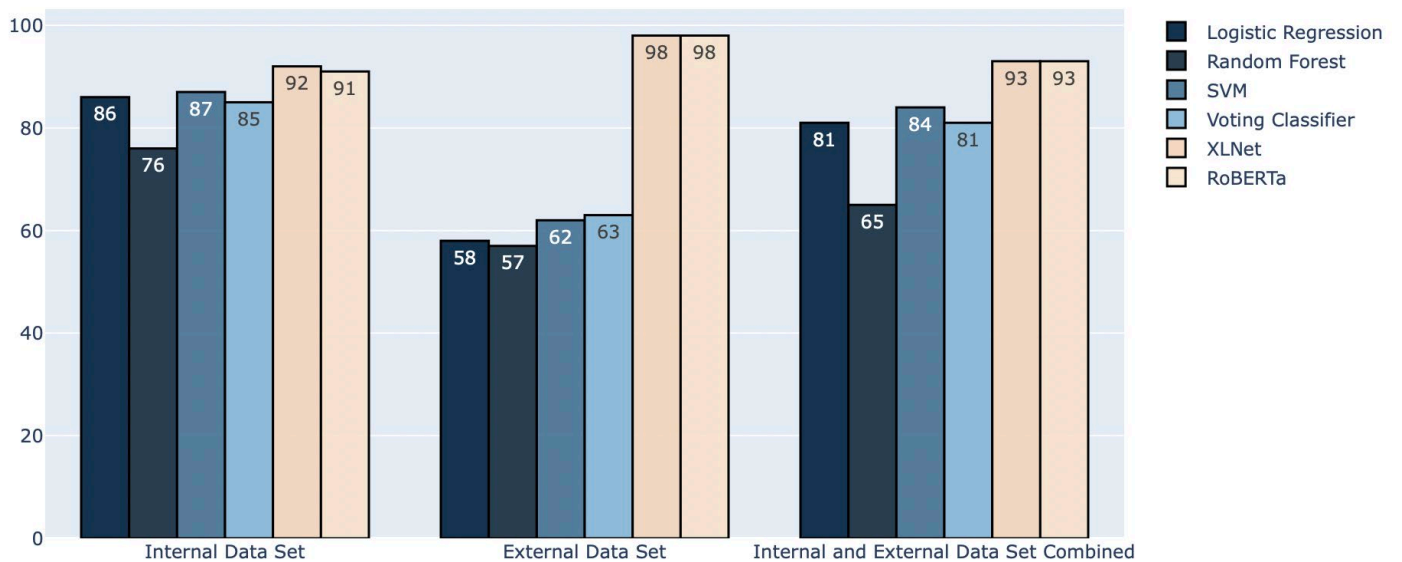


Figure 6B: Balanced Accuracy for the Six Models Trained on *Only the Internal Data Set*, *Only the External Data Set*, and the *Combined Data Set*

Source: Author's creation



XLNet trained on only an external data set offers the lowest (1%) false positive rate and the highest (98%) balanced accuracy. However, if my partner company prefers to use models trained on their proprietary data set, RoBERTa offers the lowest false positive rate of 5%,

whereas XLNet offers the highest balanced accuracy of 92%. I next discuss the implications of my research for managers and academics. I conclude with the limitations of my research and the future research they merit.

Implications for Managers

The foremost contribution is in solving a managerial problem that is becoming pervasive. Managers may use ML classifiers and conduct a first-level triage to separate ham from spam. However, they cannot rely on commercial classifiers not trained on company-specific and/or problem-specific training data. I save managers' resources by building several ML classifiers and informing them which classifiers are worth building and testing. Specifically, I report that SVM outperforms the other 26 TF-IDF-based classifiers, with a. Perhaps surprisingly, the BERT-based classifiers do not perform better than TF-IDF classifiers, suggesting that greater sophistication does not necessarily mean superior performance. However, XLNet and RoBERTa perform better than TF-IDF- and BERT-based SVM. Additionally, I assess the models' performance on multiple metrics, providing managers with a holistic assessment of which models to pursue.

My partner company has provided my software program to its customer service vendor, who is expected to apply it in my partner's production environment. After (and if) the vendor realizes the benefits of my program, it may consider applying it to the test environment of other clients. Following recent research (Hovy, Melumad, and Inman 2021; Jededi et al. 2021; Warren et al. 2021), I publish my program—the Python source code files for all classifiers I report in the manuscript—through the journal's website so that managers at other companies and researchers can use and modify the program and test its value. Once implemented, I expect the classifiers to save customer service agent's time and effort, which they may invest in core business activities.

Implications for Researchers

I offer two contributions to academics. First, research at the intersection of business and machine learning is receiving growing interest (Berger et al. 2020; Padmanabhan et al. 2022; Netzer, Lemaire, and Herzenstein 2019). Researchers have thus far mined/analyzed text generated by consumers (e.g., Mishra, Mishra, and Rathee 2019; Moore and McFerran 2017) and/or companies (e.g., Chang et al. 2021; Li, Packard, and Berger 2020; Packard and Berger 2021; Netzer et al. 2012). Such mining has helped academics extract psychology-determined and linguistically manifest features and/or topics. My research differs from this stream in both aim and method. I aim to save my partner company's operational costs (Fang, Gao, and Hu 2021) and its employees the frustration and subjectivity of classification. And I do so by automating the company's classification of incoming customer messages into spam versus ham. Because I aim to predict and not explain, I differ by using machine learning as the appropriate method.

Second, I contribute by applying machine learning models to a business problem. Specifically, I use a real data set to train 27 machine learning classifiers and identify arguably the most accurate one. Further, I validate the trade-off in the accuracy and the generalizability when machine learning models are trained on exclusively internal data set, exclusively external data set, and a combination of the two. Further, I built TF-IDF-based- and transformer-based classifiers. On the one hand, the TF-IDF-based classifiers often consume fewer computing resources than their transformer counterparts. On the other hand, transformer models are pretrained and thus offer a unique benefit (Alantari, Currim, Deng, and Singh 2022). Therefore, academics may consider trading off the pretraining benefits with resource costs.

Limitations and Future Research

I see two limitations of my research. First, I claim that my spam classifier can help a company save costs and lower employees' frustration in manually labeling incoming messages. But I provide no evidence for my claims. Unfortunately, while my partner company would likely adopt my classifier, I do not have the data (as of yet) on whether and how much the company will save. Similarly, while I would prefer to survey the service agents (Altman et al. 2021), I do not know when exactly the outsourcing customer service vendor will implement my classifier in production. The current pandemic-caused retail and supply chain disruption has added to the uncertainty. Future research may thus consider measuring the performance effects of companies adopting a spam classifier and machine learning solutions.

Second, I classify incoming customer messages that my partner company received as emails or fill out a "contact us" form. These messages are relatively long and thus lend themselves well to training a machine learning classifier. However, customers now interact with companies on messaging apps and social media platforms. These interactions tend to be shorter. Unfortunately for me, my partner company—at least, as of now—does not receive enough short messages from customers, and thus, my research is limited to relatively long messages. However, I foresee the possibility of spam appearing in customer messages sent over messaging apps and social media platforms. Future research may consider addressing the limitations of my research by applying machine learning classifiers to shorter messages.

REFERENCES

Altman, Daniel, Galit B. Yom-Tov, Marcelo Olivares, Shelly Ashtar, and Anat Rafaeli (2021), "Do Customer Emotions Affect Agent Speed? An Empirical Study of Emotional Load in Online Customer Contact Centers," *Manufacturing & Service Operations Management*, 23 (4), 854-875.

Berger, Jonah, Grant Packard, Reihane Boghrati, Ming Hsu, Ashlee Humphreys, Andrea Luangrath, Sarah Moore, Gideon Nave, Christopher Olivola, and Matthew Rocklage (2022), "Wisdom from Words: Marketing Insights from Text," *Marketing Letters*, Forthcoming.

- Berger, Jonah, Ashlee Humphreys, Stephan Ludwig, Wendy W Moe, Oded Netzer, and David A Schweidel (2020), “Uniting the Tribes: Using Text for Marketing Insight,” *Journal of Marketing*, 84(1), 1–25.
- Blanzieri, E., & Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1), 63-92.
- Bhopale, A. P., & Tiwari, A. (2021, March). An Application of Transfer Learning: Fine-Tuning BERT for Spam Email Classification. In *International Conference on Machine Learning and Big Data Analytics* (pp. 67-77). Springer, Cham.
- Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003), “Latent Dirichlet Allocation,” *Journal of Machine Learning Research*, 3(Jan), 993-1022.
- Boghtrati, Reihane, and Jonah Berger, “Quantifying Gender Bias in Consumer Culture,” available at <https://arxiv.org/ftp/arxiv/papers/2201/2201.03173.pdf>.
- Chang, Xiangyu, Yinghui Huang, Mei Li, Xin Bo, and Subodha Kumar (2021), “Efficient Detection of Environmental Violators: A Big Data Approach,” *Production and Operations Management*, 30 (5), 1246-1270.
- Dada, E. G., Bassi, J. S., Chiroma, H., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6), e01802.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018), “Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*.
- Dodds, Peter Sheridan and Christopher M Danforth (2010), “Measuring the Happiness of Large-Scale Written Expression: Songs, Blogs, and Presidents,” *Journal of happiness studies*, 11(4), 441–56.
- Fang, Xiao, Yuanyuan Gao, and Paul J. Hu (2021), “A prescriptive analytics method for cost reduction in clinical decision making,” *MIS Quarterly*, 45(1), 83-115.
- Guo, Xiaojia, Yael Grushka-Cockayne, and Bert De Reyck (2021), “Forecasting Airport Transfer Passenger Flow Using Real-time Data and Machine Learning.” *Manufacturing & Service Operations Management* (2021).
- Hartmann, J., & Netzer, O. (2023). Natural language processing in marketing. In *Artificial intelligence in marketing* (Vol. 20, pp. 191-215). Emerald Publishing Limited.
- Hovy, Dirk, Shiri Melumad, and J. Jeffrey Inman (2021), “Wordify: A Tool for Discovering and Differentiating Consumer Vocabularies,” *Journal of Consumer Research*, Forthcoming.
- Hu, Kejia, Gad Allon, and Achal Bassamboo (2021), “Understanding customer retrials in call centers: Preferences for service quality and service speed,” *Manufacturing & Service Operations Management*, 24(2), 1002-1020.
- Hutto, Clayton J and Eric Gilbert (2014), “Vader: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” in *Eighth International AAAI Conference on Weblogs and Social Media*.
- Lee, P. S., Chakraborty, I., & Banerjee, S. (2023). Artificial intelligence applications to customer feedback research: A review. *Artificial Intelligence in Marketing*, 169-190.
- Liu, X. (2023). Deep learning in marketing: a review and research agenda. *Artificial Intelligence in Marketing*, 239-271.
- Jedidi, Kamel, Bernd H. Schmitt, Malek Ben Sliman, and Yanyan Li (2021), “R2M Index 1.0: Assessing

- the Practical Relevance of Academic Marketing Articles,” *Journal of Marketing*, Forthcoming.
- Johnson, Joseph (2021), “Spam: Share of Global Email Traffic, 2014-2021), Statista, January 4, 2021.
- Li, Yang, Grant Packard, and Jonah Berger (202), “Conversational Dynamics: When Does Employee Language Matter?” Working Paper.
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019), “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692* (2019).
- Mejia, Jorge, Shawn Mankad, and Anandasivam Gopal (2021), “Service Quality Using Text Mining: Measurement and Consequences.” *Manufacturing & Service Operations Management*, 23 (6), 1354-1372.
- Mishra, Arul, Himanshu Mishra, and Shelly Rathee (2019), “Examining the Presence of Gender Bias in Customer Reviews Using Word Embedding.” *arXiv preprint arXiv:1902.00496* (2019).
- Moore, Sarah G and Brent McFerran (2017), “She Said, She Said: Differential Interpersonal Similarities Predict Unique Linguistic Mimicry in Online Word of Mouth,” *Journal of the Association for Consumer Research*, 2(2), 229–45.
- Netzer, Oded, Ronen Feldman, Jacob Goldenberg, and Moshe Fresko (2012), “Mine Your Own Business: Market-Structure Surveillance through Text Mining,” *Marketing Science*, 31(3), 521–43.
- Netzer, Oded, Alain Lemaire, and Michal Herzenstein (2019), “When Words Sweat: Identifying Signals for Loan Default in the Text of Loan Applications,” *Journal of Marketing Research*, 56(6), 960–80.
- Nisar, N., Rakesh, N., & Chhabra, M. (2021). Review on Email Spam Filtering Techniques. *International Journal of Performability Engineering*, 17(2).
- Packard, Grant, and Jonah Berger (2021), “How Concrete Language Shapes Customer Satisfaction,” *Journal of Consumer Research*, 47 (5), 787-806.
- Packard, Grant, and Jonah Berger (2020), “Thinking of you: How second-person pronouns shape cultural success.” *Psychological Science*, 31 (4), 397-407.
- Padmanabhan, Balaji, Nachiketa Sahoo, and Andrew Burton-Jones (2022), “Machine Learning in Information Systems Research,” *Management Information Systems Quarterly*, 46 (1), iii-xix.
- Packard, Grant, Sarah G Moore, and Brent McFerran (2018), “(I’m) Happy to Help (You): The Impact of Personal Pronoun Use in Customer-Firm Interactions,” *Journal of Marketing Research*, 55(4), 541–55.
- Pennebaker, James W., Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. *The development and psychometric properties of LIWC2015*. 2015.
- Puranam, Dinesh, Vrinda Kadiyali, and Vishal Narayan (2021), “The Impact of Increase in Minimum Wages on Consumer Perceptions of Service: A Transformer Model of Online Restaurant Reviews,” *Marketing Science*, 40 (5), 985-1004.
- Rocklage, Matthew D, Derek D Rucker, and Loran F Nordgren (2018), “The Evaluative Lexicon 2.0: The Measurement of Emotionality, Extremity, and Valence in Language,” *Behavior Research Methods*, 50(4), 1327–44.
- Samorani, Michele, Shannon L. Harris, Linda Goler Blount, Haibing Lu, and Michael A. Santoro (2021), “Overbooked and Overlooked: Machine Learning and Racial Bias in Medical Appointment Scheduling,” *Manufacturing & Service Operations Management*, Forthcoming.

Smith, Dale (2020), “6 Clever Tricks to Minimize Regret, Frustration, and Spam,” <https://www.cnet.com/tech/services-and-software/6-clever-gmail-tricks-to-minimize-regret-frustration-and-spam/>, January 4, 2021.

Srinivasan, Sriram, Vinayakumar Ravi, Mamoun Alazab, Simran Ketha, Ala’M. Al-Zoubi, and Soman Kotti Padannayil (2021), “Spam Emails Detection Based on Distributed Word Embedding with Deep Learning.” In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*, 161-189. Springer, Cham, 2021.

Southeastern Technical (2020), “Why is It So Difficult to Detect Phishing Emails,” <https://www.setechnical.net/data-security/why-is-it-so-difficult-to-detect-phishing-emails/>, January 4, 2021.

Sun, Zhankun, Nilay Tanik Argon, and Serhan Ziya (2022), “When to Triage in Service Systems with Hidden Customer Class Identities?” *Production and Operations Management*, 31 (1), 172-193.

Warren, Nooshin L., Matthew Farmer, Tianyu Gu, and Caleb Warren (2021), “Marketing Ideas: How to Write Research Articles that Readers Understand and Cite,” *Journal of Marketing*, 85 (5), 42-57.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le (2019), “XLNet: Generalized Autoregressive Pretraining for Language Understanding,” *Advances in Neural Information Processing Systems*, 32.

Yaseen, Qussai (2021), “Spam Email Detection Using Deep Learning Techniques,” *Procedia Computer Science*, 184, 853-858.

Zhu, Xiaodan, Anh Ninh, Hui Zhao, and Zhenming Liu (2021), “Demand Forecasting with Supply-chain Information and Machine Learning: Evidence in the Pharmaceutical Industry,” *Production and Operations Management*, 30 (9), 3231-3252.

Classifying an Incoming Customer Message into Spam Versus Ham

Web Appendix

Web Appendix A: Accuracy Rates from BERT-based Classifiers Trained on *Only the Internal Data Set*

Table A1: Confusion Matrix for the BERT-based *Logistic Regression* Classifier Trained on *Only the Internal Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.78%)	118,265 (True Negatives) 84.79%	21,212 (False Positives) 15.20%
<i>Spam</i>	4,626 (3.22%)	535 (False Negatives) 11.56%	4,091 (True Positives) 88.43%
Total	144,103	118,800	25,303

Table A2: Classification Report for the BERT-based *Logistic Regression* Classifier Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.85	.92
1 (Spam)	.16	.88	.27
Accuracy	.84		
Balanced Accuracy	.86		

Figure A1: ROC Curve for the *Logistic Regression* Classifier Trained on *Only the Internal Data Set*

Source: Author's creation

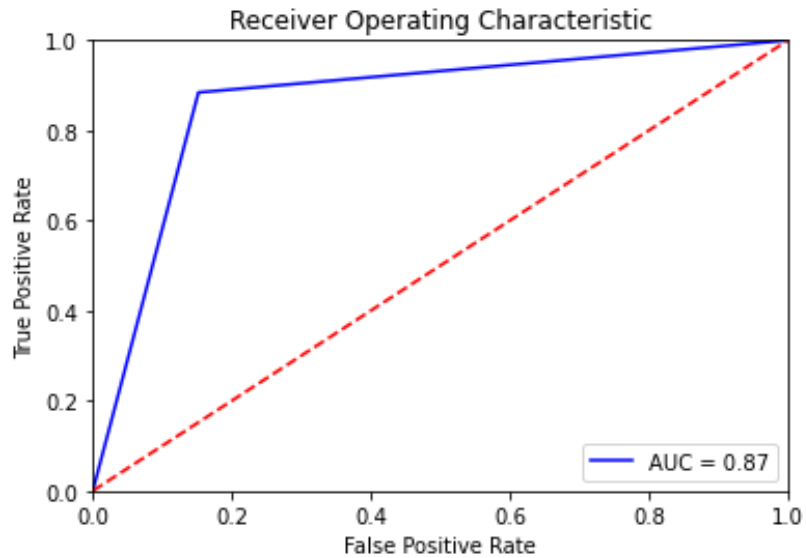


Table A3: Confusion Matrix for the *Random Forest Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Random Forest Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.8%)	129,528 (True Negatives) 92.86%	9,949 (False Positives) 7.14%
<i>Spam</i>	4,626 (3.2%)	1,820 (False Negatives) 39.34%	2,806 (True Positives) 60.65%
Total	144,103	131,348	12,755

Table A4: Classification Report for the *Random Forest Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.93	.96
1 (Spam)	.22	.61	.32
Accuracy	.91		
Balanced Accuracy	.76		

Figure A2: ROC Curve for the *Random Forest Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

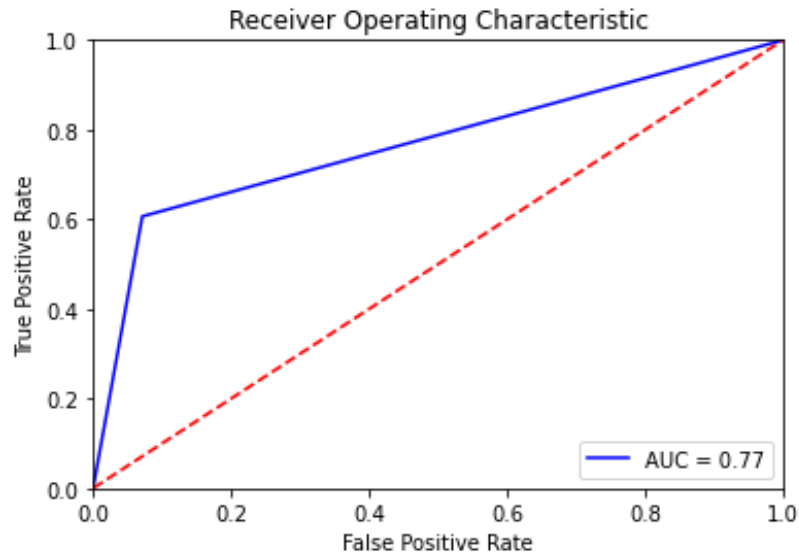


Table A5: Confusion Matrix for the *SVM Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the SVM Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.8%)	120,670 (True Negatives) 86.51%	18,807 (False Positives) 13.49%
<i>Spam</i>	4,626 (3.2%)	606 (False Negatives) 13.10%	4,020 (True Positives) 86.90%
Total	144,103	121,276	22,827

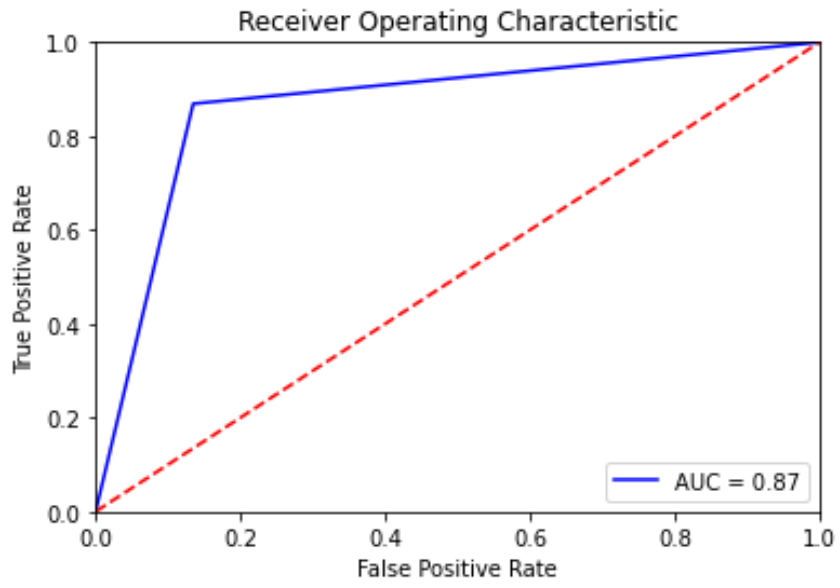
Table A6: Classification Report for the *SVM Classifier* Trained on *Only the Internal Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.87	.93
1 (Spam)	.18	.87	.29
Accuracy	.86		
Balanced Accuracy	.87		

Figure A3: ROC Curve for the *SVM Classifier* Trained Using *Only the Internal Data Set*

Source: Author's creation



Web Appendix B: Accuracy Rates from BERT-based Classifiers Trained on *Only the External Data Set*

Table B1: Confusion Matrix for the BERT-based *Logistic Regression* Classifier Trained on *Only the External Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	142,277 (96.06%)	71,905 (True Negatives) 50.53%	70,372 (False Positives) 49.47%
<i>Spam</i>	5,826 (3.94%)	1,975 (False Negatives) 33.89%	3,851 (True Positives) 66.11%
Total	148,103	73,880	74,223

Table B2: Classification Report for the BERT-based *Logistic Regression* Classifier Trained on *Only the External Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.97	.51	.67
1 (Spam)	.05	.66	.10
Accuracy	.51		
Balanced Accuracy	.58		

Figure B1: ROC Curve for the *Logistic Regression* Classifier Trained on *Only the External Data Set*

Source: Author's creation

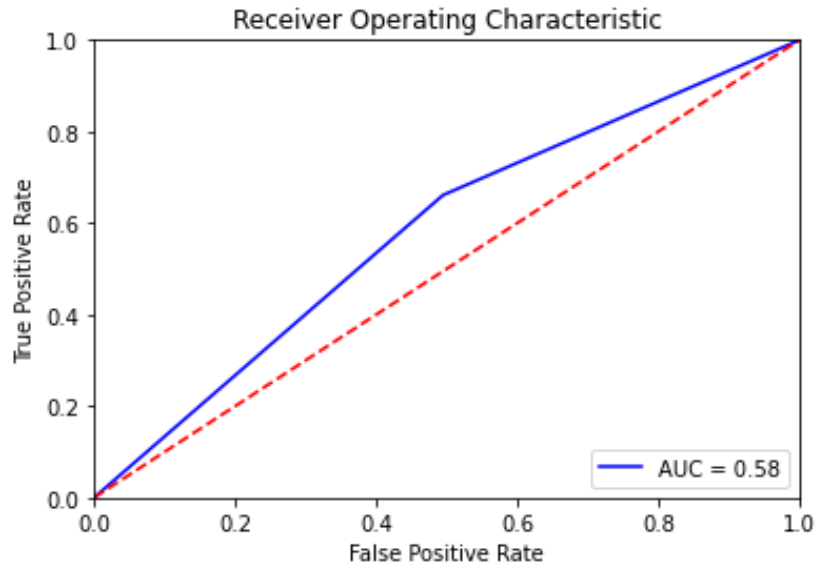


Table B3: Confusion Matrix for the *Random Forest* Classifier Trained on *Only the External Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Random Forest Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	142,277 (96.06%)	126,278 (True Negatives) 88.75%	15,999 (False Positives) 11.25%
<i>Spam</i>	5,826 (3.94%)	4,346 (False Negatives) 74.59%	1,480 (True Positives) 25.41%
Total	148,103	130,624	17,479

Table B4: Classification Report for the Random Forest Classifier Trained on *Only the External Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.97	.89	.93
1 (Spam)	.08	.25	.13
Accuracy	.86		
Balanced Accuracy	.57		

Figure B2: ROC Curve for the Random Forest Classifier Trained on *Only the External Data Set*

Source: Author's creation

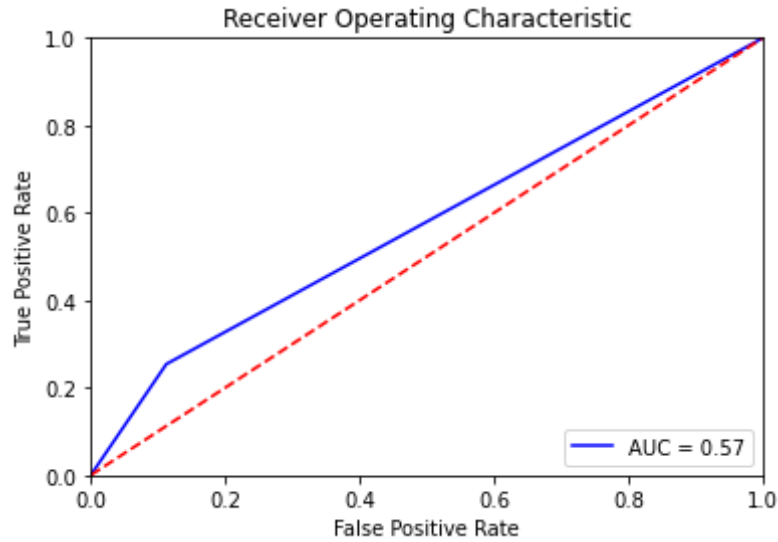


Table B5: Confusion Matrix for the SVM Classifier Trained on *Only the External Data Set*

Source: Author’s creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	142,277 (96.06%)	92,922 (True Negatives) 65.31%	49,355 (False Positives) 34.69%
<i>Spam</i>	5,826 (3.94%)	2,253 (False Negatives) 38.67%	3,573 (True Positives) 61.33%
Total	148,103	95,175	52,928

Table B6: Classification Report for the SVM Classifier Trained on *Only the External Data Set*

Source: Author’s creation

Class	Precision	Recall	F1 Score
0 (Ham)	.98	.65	.78
1 (Spam)	.07	.61	.12
Accuracy	.65		
Balanced Accuracy	.62		

Figure B3: ROC Curve for the SVM Classifier Trained on *Only the External Data Set*

Source: Author’s creation

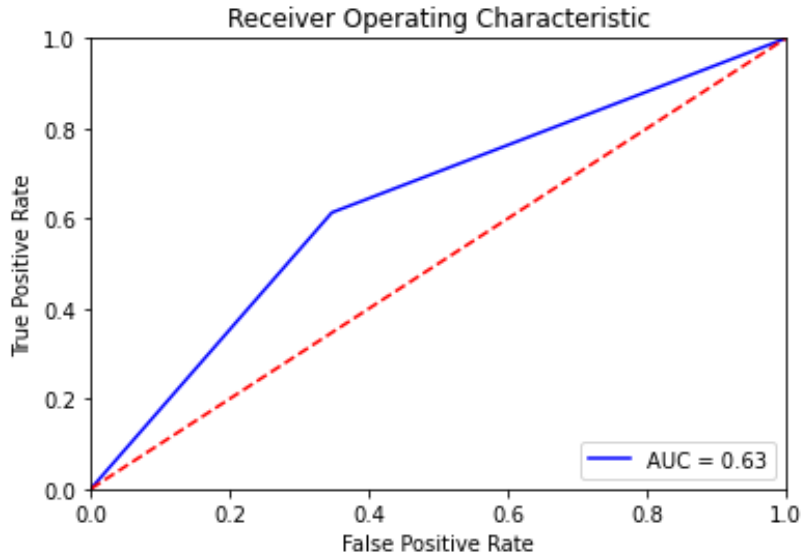


Table B7: Confusion Matrix for the Voting Classifier Trained on *Only the External Data Set*

Source: Author’s creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	142,277 (96.78%)	95,408 (True Negatives) 88.31%	46,869 (False Positives) 11.68%
<i>Spam</i>	5,826 (3.22%)	2,480 (False Negatives) 18.54%	3,346 (True Positives) 81.45%
Total	148,103	97,888	50,215

Table B8: Classification Report for the Voting Classifier Trained on *Only the External Data Set*

Source: Author’s creation

Class	Precision	Recall	F1 Score
0 (Ham)	.97	.67	.79
1 (Spam)	.07	.57	.12
Accuracy	.67		
Balanced Accuracy	.63		

Figure B4: ROC Curve for the Voting Classifier Trained on *Only the External Data Set*

Source: Author’s creation

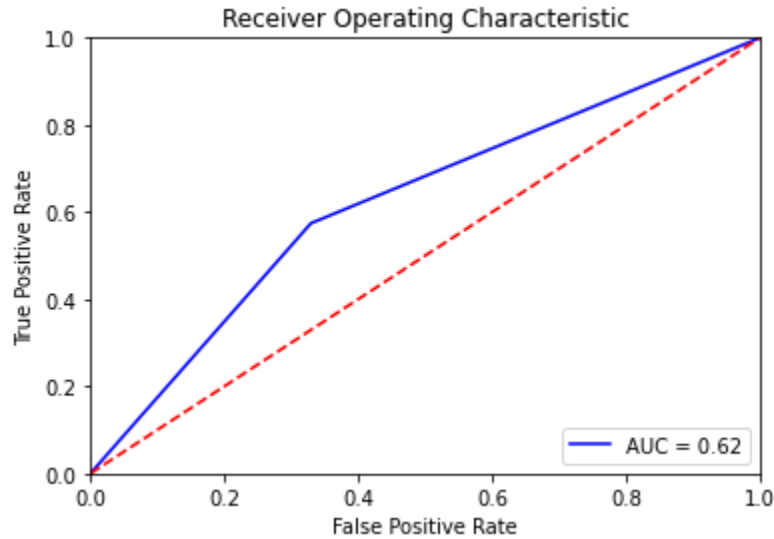


Table B9: Confusion Matrix for XLNet Trained on *Only the External Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the XLNet Classifier on External Dataset		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	1458 (72%)	1448 (True Negatives) 99.31%	10 (False Positives) 0.69%
<i>Spam</i>	575 (28%)	17 (False Negatives) 2.96%	558 (True Positives) 97.04%
Total	2,033	1,465	568

Table B10: Classification Report for XLNet Trained on *Only the External Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.99	.99
1 (Spam)	.98	.97	.97
Accuracy	.99		
Balanced Accuracy	.98		

Figure B5: ROC Curve for the XLNet Trained on *Only the External Data Set*

Source: Author's creation

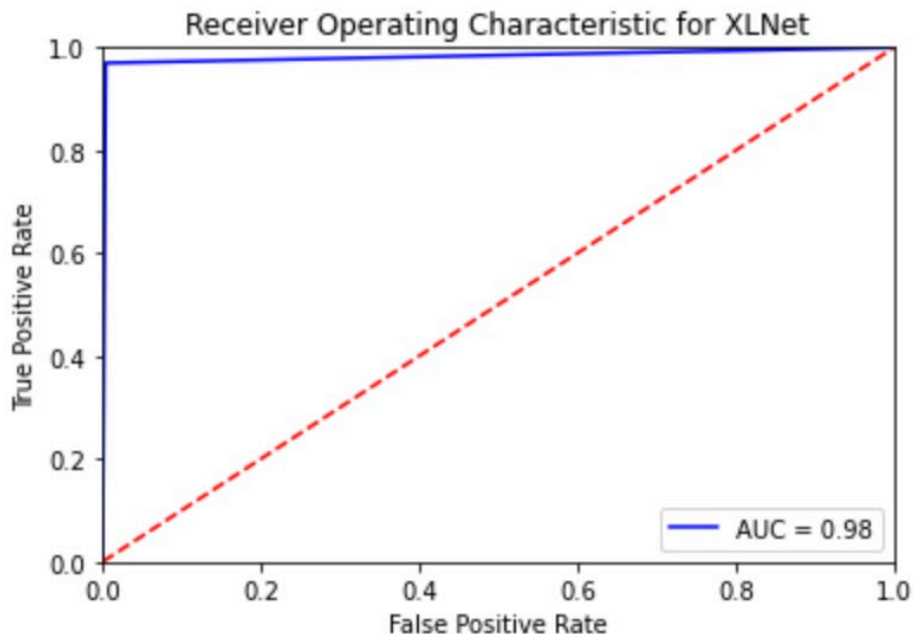


Table B11: Confusion Matrix for RoBERTa Trained on *Only the External Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the XLNet Classifier on External Dataset		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	1465 (72%)	1451 (True Negatives) 99.04%	7 (False Positives) 0.96%
<i>Spam</i>	575 (28%)	17 (False Negatives) 2.96%	558 (True Positives) 97.04%
Total	2,033	1468	565

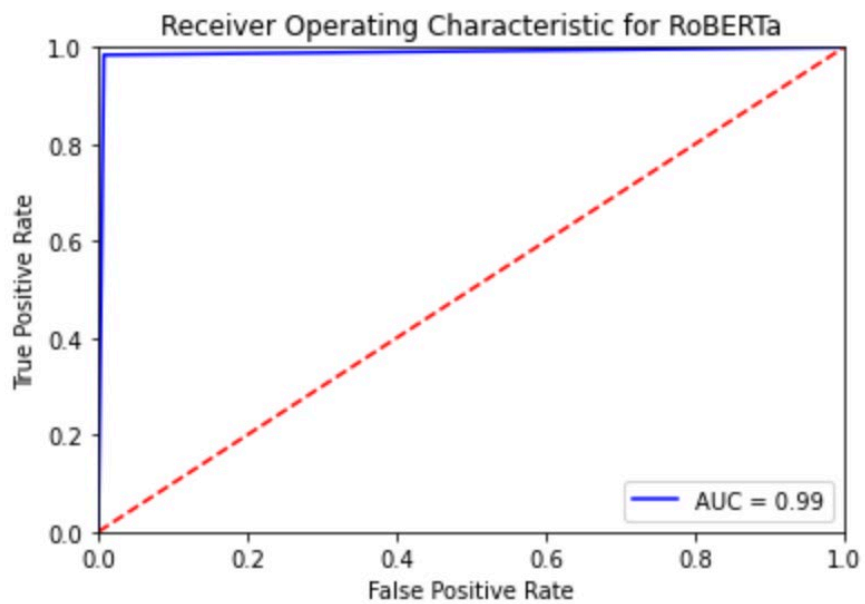
Table B12: Classification Report for RoBERTa Trained on *Only the External Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.99	.99
1 (Spam)	.99	.97	.98
Accuracy	.99		
Balanced Accuracy	.98		

Figure B6: ROC Curve for the *RoBERTa* Trained on *Only the External Data Set*

Source: Author's creation



Web Appendix C: Accuracy Rates from BERT-based Classifiers Trained on the *Combined* Internal and External Data Set

Table C1: Confusion Matrix for the BERT-based *Logistic Regression* Classifier Trained on the *Combined* Data Set

Source: Author's creation

Classified by the Case Owner	Classified by the Logistic Regression Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.78%)	115,829 (True Negatives) 83.04%	23,648 (False Positives) 16.96%
<i>Spam</i>	4,626 (3.22%)	998 (False Negatives) 21.57%	3,628 (True Positives) 78.43%
Total	144,103	116,827	27,276

Table C2: Classification Report for the BERT-based *Logistic Regression* Classifier Trained on the *Combined* Data Set

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.83	.90
1 (Spam)	.13	.78	.23
Accuracy	.83		
Balanced Accuracy	.81		

Figure C1: ROC Curve for the *Logistic Regression* Classifier Trained on the *Combined* Data Set

Source: Author's creation

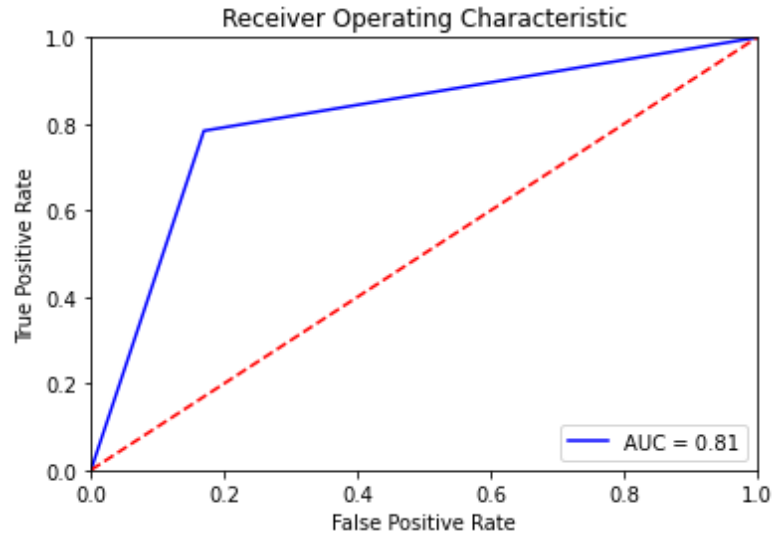


Table C3: Confusion Matrix for the *Random Forest* Classifier Trained on the *Combined* Data Set

Source: Author's creation

Classified by the Case Owner	Classified by the Random Forest Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.78%)	133,232 (True Negatives) 95.52%	6,245 (False Positives) 4.48%
<i>Spam</i>	4,626 (3.22%)	3,018 (False Negatives) 65.23%	1,608 (True Positives) 34.76%
Total	144,103	136,250	7,853

Table C4: Classification Report for the *Random Forest* Classifier Trained on the *Combined* Data Set

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.98	.96	.97
1 (Spam)	.20	.35	.26
Accuracy	.93		
Balanced Accuracy	.65		

Figure C2: ROC Curve for the *Random Forest* Classifier Trained on the *Combined* Data Set

Source: Author's creation

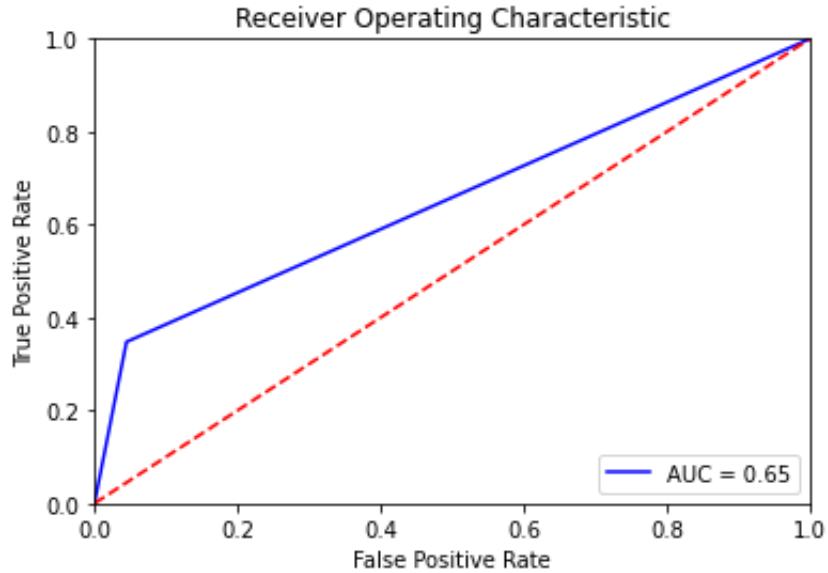


Table C5: Confusion Matrix for the *SVM Classifier* Trained on the *Combined Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the SVM Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.78%)	120,129 (True Negatives) 86.12%	19,348 (False Positives) 13.87%
<i>Spam</i>	4,626 (3.21%)	839 (False Negatives) 18.13%	3,787 (True Positives) 81.86%
Total	144,103	120,968	23,135

Table C6: Classification Report for the *SVM Classifier* Trained on the *Combined Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.86	.92
1 (Spam)	.16	.82	.27
Accuracy	.86		
Balanced Accuracy	.84		

Figure C3: ROC Curve for the *SVM Classifier* Trained on the *Combined Data Set*

Source: Author's creation

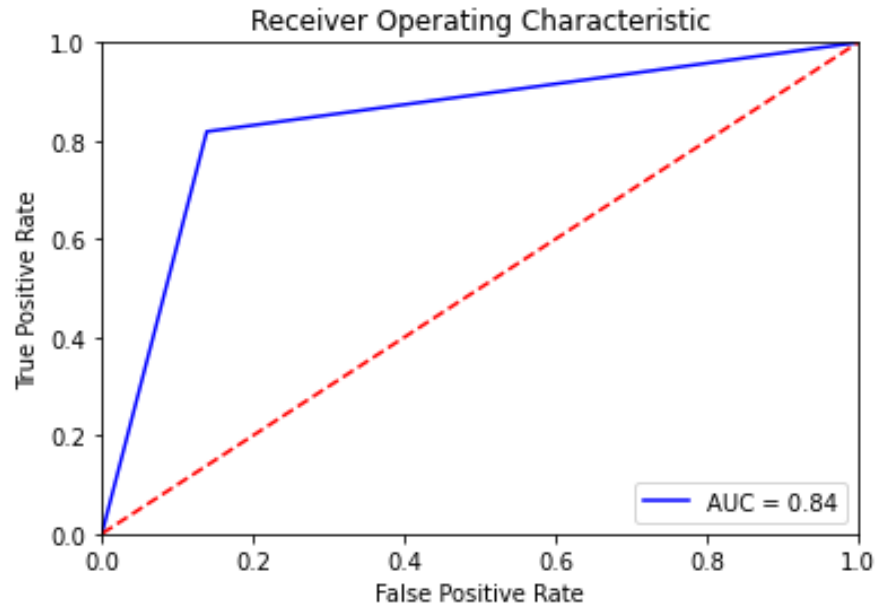


Table C7: Confusion Matrix for the *Voting Classifier* Trained on the *Combined Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the Voting Classifier		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	139,477 (96.78%)	122,468 (True Negatives) 88.31%	17,009 (False Positives) 11.68%
<i>Spam</i>	4,626 (3.22%)	1,170 (False Negatives) 18.54%	3,456 (True Positives) 81.45%
Total	144,103	123,638	20,465

Table C8: Classification Report for the *Voting Classifier* Trained on the *Combined Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.99	.88	.93
1 (Spam)	.17	.75	.28
Accuracy	.87		
Balanced Accuracy	.81		

Figure C4: ROC Curve for the *Voting Classifier* Trained on the *Combined Data Set*

Source: Author's creation

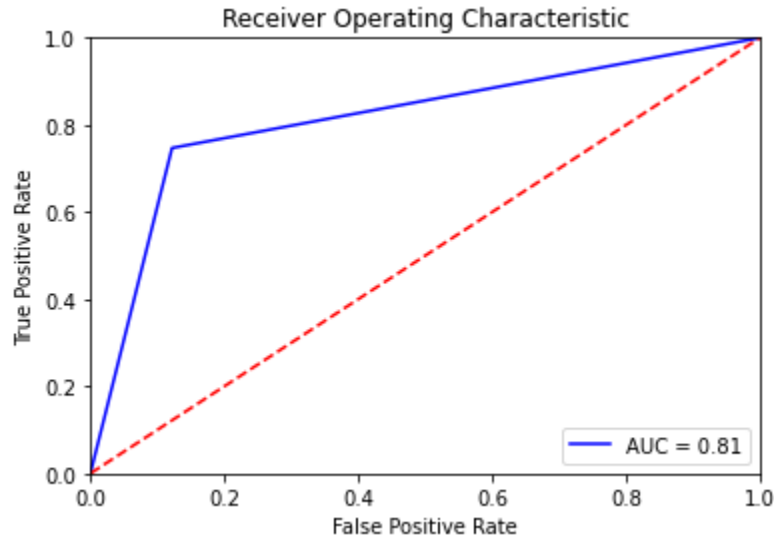


Table C9: Confusion Matrix for XLNet Trained on the Combined Data Set

Source: Author's creation

Classified by the Case Owner	Classified by the XLNet Classifier on Internal and External Dataset		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	4926 (73%)	4692 (True Negatives) 95.24%	234 (False Positives) 4.76%
<i>Spam</i>	1763 (27%)	136 (False Negatives) 7.72%	1627 (True Positives) 92.28%
Total	6,689	4,828	1861

Table C10: Classification Report for XLNet Trained on the Combined Data Set

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.97	.95	.96
1 (Spam)	.87	.92	.90
Accuracy	.94		
Balanced Accuracy	.93		

Figure C5: ROC Curve for the XLNet Trained on the Combined Data Set

Source: Author's creation

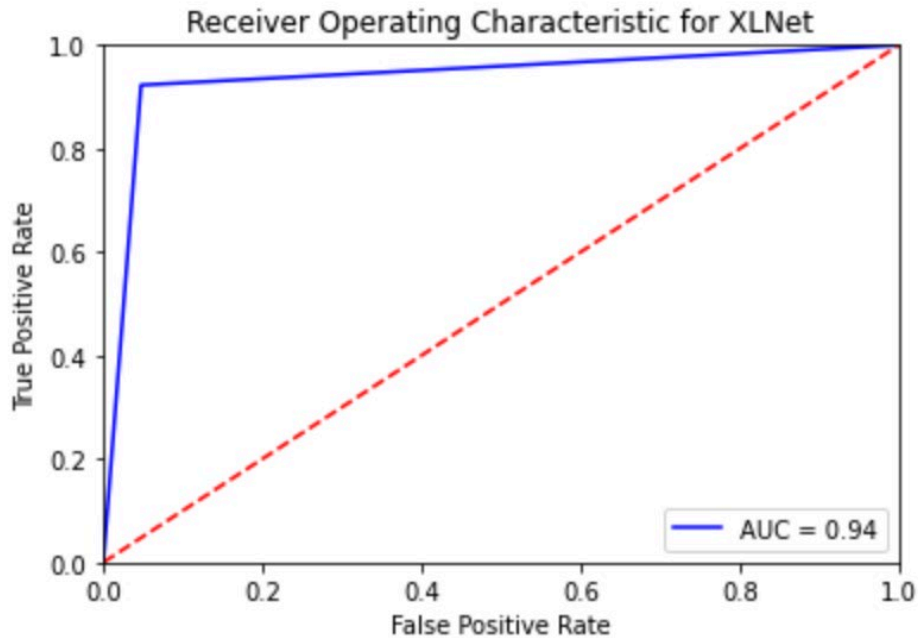


Table C11: Confusion Matrix for *RoBERTa* Trained on the *Combined Data Set*

Source: Author's creation

Classified by the Case Owner	Classified by the XLNet Classifier on External Dataset		
		<i>Ham</i>	<i>Spam</i>
<i>Ham</i>	4,926 (73%)	4,703 (True Negatives) 95.47%	223 (False Positives) 4.53%
<i>Spam</i>	1763 (27%)	139 (False Negatives) 7.88%	1,624 (True Positives) 92.11%
Total	6,689	4,842	1,847

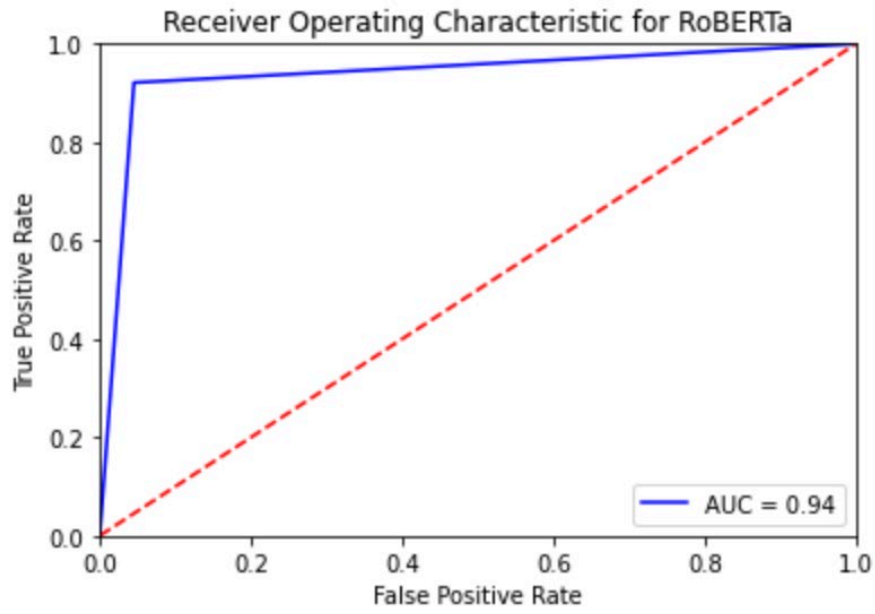
Table C12: Classification Report for *RoBERTa* Trained on the *Combined Data Set*

Source: Author's creation

Class	Precision	Recall	F1 Score
0 (Ham)	.97	.95	.96
1 (Spam)	.88	.92	.90
Accuracy	.94		
Balanced Accuracy	.93		

Figure C6: ROC Curve for the *RoBERTa* Trained on the *Combined Data Set*

Source: Author's creation



Web Appendix D

On Spam: LDA, KeyBERT + LDA, KeyBERT + Guided LDA

Spam Data Set: LDA

I built an LDA model, setting the #topics to the following values: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 50. That is, I built 21 LDA models. For each LDA model, I computed coherence (see Table D1A and Figure D1A) and perplexity (see Table D1B and Figure D1B). The tables and figures indicate that 12 is the optimal number of topics.

Table D1A: Spam Data Set: Coherence Scores of LDA Models

Source: Author's creation

Number of Topics	Coherence Score
2	0.45
3	0.38
4	0.38
5	0.46
6	0.53
7	0.49
8	0.52
9	0.44
10	0.46
11	0.56
12	0.57
13	0.46
14	0.49
15	0.53
16	0.50
17	0.52
18	0.51
19	0.50
20	0.53
25	0.56
50	0.51

Figure D1A: Spam Data Set: Coherence Scores of LDA Models

Source: Author's creation

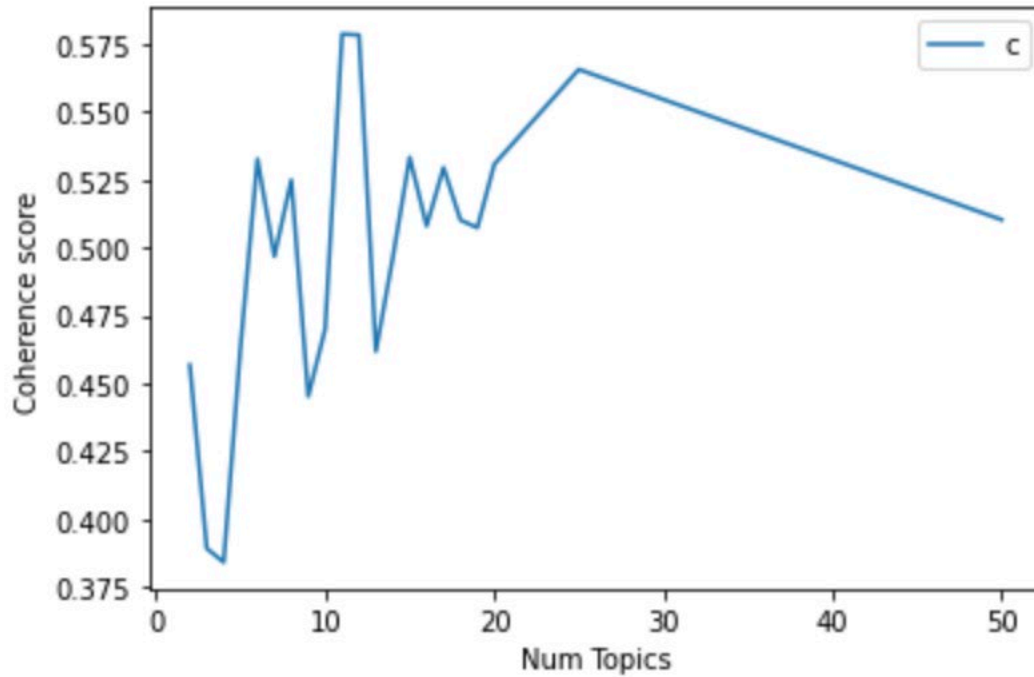


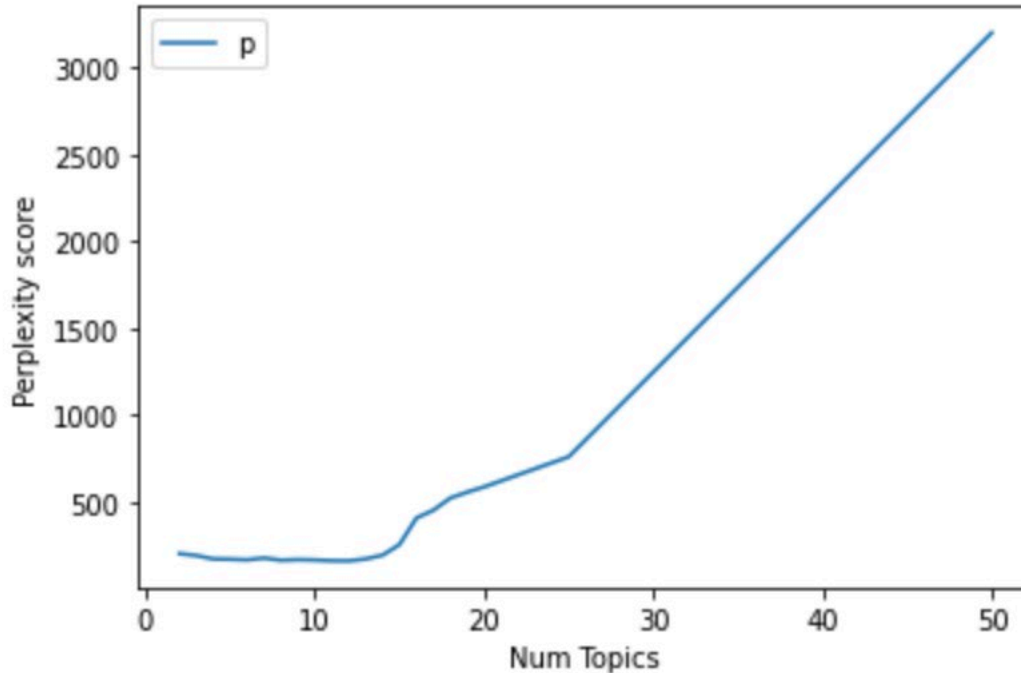
Table D1B: Spam Data Set: Perplexity Scores of LDA Models

Source: Author's creation

Number of Topics	Perplexity Score
2	200.90
3	190.09
4	170.30
5	168.87
6	165.34
7	175.77
8	162.87
9	166.57
10	164.41
11	160.26
12	159.45
13	170.43
14	192.132
15	253.38
16	406.21
17	451.19
18	521.18
19	555.34
20	585.51
25	758.98
50	3202.06

Figure D1B: Spam Data Set: Perplexity Scores of LDA Models

Source: Author's creation

**Figure D1C: Spam Data Set: LDA With #Topics = 12**

Source: Author's creation

Topic 0: new email view today franklin free county order products read
 Topic 1: shoe carnival order email image work link contact privacy shared
 Topic 2: receive email ram follow messages dear read spam message click
 Topic 3: products best china shoes company dear email contact tel ltd
 Topic 4: intended information said message mail recipient confidential office use emai
 Topic 5: email office message contact account address click return emails mail
 Topic 6: business website help get best google time company services email
 Topic 7: und wir email werden anfragen sie mansfield service privacy diese
 Topic 8: aircraft hours total aviation new view jet learn number offered
 Topic 9: shoecarnival shoe important carnival text wrote none font inherit height
 Topic 10: question click date con este los text last shoe por
 Topic 11: development web apps interested list email unsubscribe name app website

I struggle to provide labels to these topics. The struggle makes sense because, by definition, spam data set should not have topics that one can label without extreme difficulty.

Spam Data Set: KeyBERT + LDA

For the 5,826 processed spam emails, I ran KeyBERT to identify the top unigram and bigram storing the results in a new column. I set #topics = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 50]. I append the coherence scores and perplexity scores table and

graph next. Coherence is highest for 14 topics. However, perplexity is lowest for eight topics. I rate coherence above perplexity and proceeded with 14 topics.

Table D2A: Spam Data Set: Coherence Scores for KeyBERT + LDA Models

Source: Author's creation

Number of Topics	Coherence Score
2	0.73
3	0.72
4	0.72
5	0.74
6	0.71
7	0.72
8	0.72
9	0.72
10	0.72
11	0.72
12	0.71
13	0.71
14	0.70
15	0.71
16	0.71
17	0.71
18	0.72
19	0.73
20	0.74
25	0.74
50	0.76

Figure D2A: Spam Data Set: Coherence Scores for KeyBERT + LDA Models

Source: Author's creation

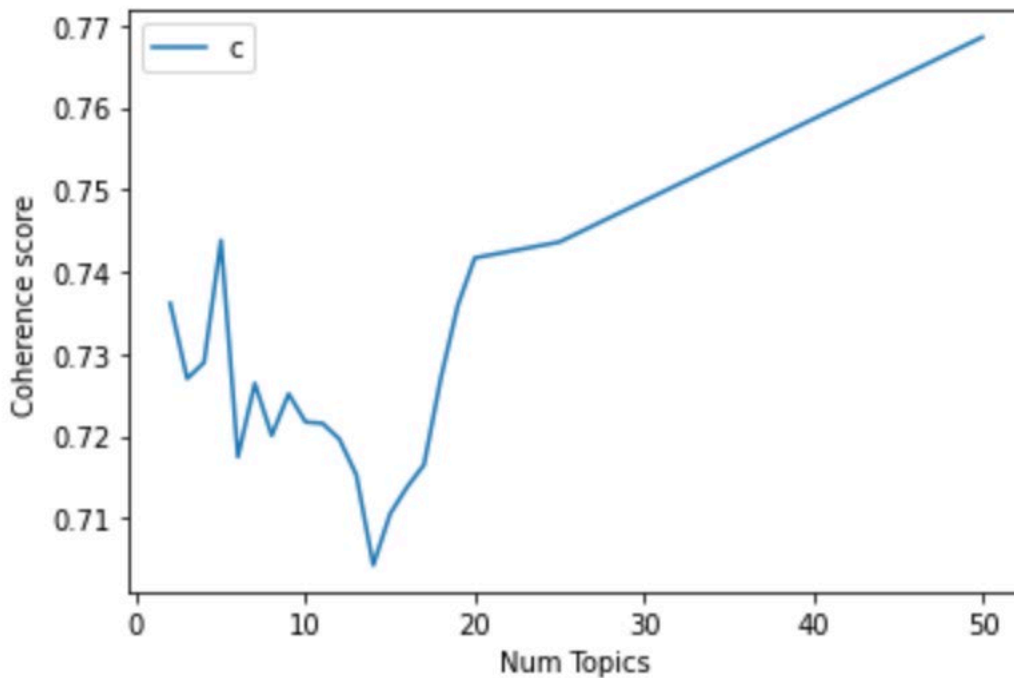


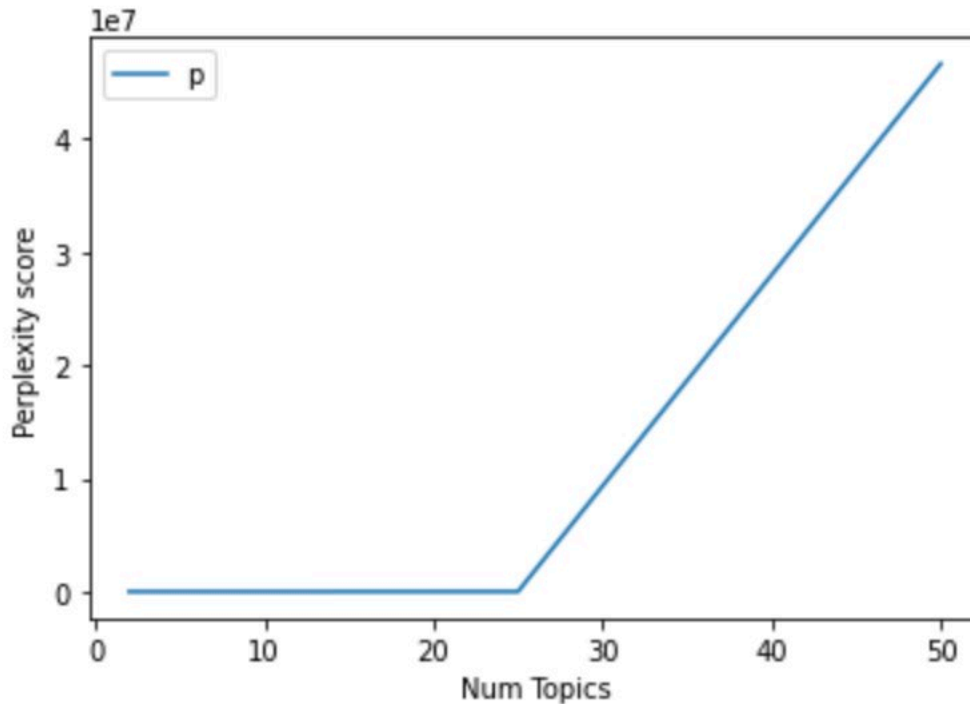
Table D2B: Spam Data Set: Perplexity Scores for KeyBERT + LDA Models

Source: Author's creation

Number of Topics	Perplexity Score
2	26.81
3	24.51
4	23.69
5	23.17
6	22.26
7	21.67
8	20.50
9	21.51
10	25.58
11	25.39
12	24.43
13	24.76
14	91.40
15	108.15
16	127.92
17	246.53
18	438.40
19	548.10
20	845.24
25	1920.17
50	46618485.57

Figure D2B: Spam Data Set: Perplexity Scores on KeyBERT + LDA Models

Source: Author's creation

**Figure D2C: Spam Data Set: KeyBERT + LDA With #Topics = 14**

Source: Author's creation

Topic 0: business web website app customerservice marketing file esports designer proposal
 Topic 1: response expect spam mail logo jet contact delivery products vape
 Topic 2: email new county shop weekly retail list view form russellville
 Topic 3: shoes market privacy manufacturer bearbeitungsnummer gespeichert statement bags customer account
 Topic 4: moved donor information mail contact notice confidentiality personal iphone wholesale
 Topic 5: shoecarnival carnival psychic wrote miranda escribió mansfield mailto date dear
 Topic 6: html doctype password domain shoes email service text holiday newsletter
 Topic 7: email inventory viewing contacting county today ceo warning exchange chat
 Topic 8: email order publicidad informamos offer account address customerservice fax apparel
 Topic 9: business aviation company seo services gulfstream online apps service industry
 Topic 10: unsubscribe nicotine safe products boots document customers sales info bags
 Topic 11: office contact assistance return emails reply returning closed monday mail
 Topic 12: message messages ram follow email invoice secure shoes office dear
 Topic 13: shoe carnival shoecarnival message perks text customerservice ww error recipient

I continue to struggle to label these 14 topics, further supporting the insight that the spam emails lack any topic.

On Ham: LDA, KeyBERT + LDA, KeyBERT + Guided LDA

Source: Author's creation

Ham Data Set: LDA

I repeated the steps for the ham data set. The coherence score is the highest when the #topics is 15, but the perplexity score is the lowest when #topics is 12. I chose coherence over perplexity and thus proceeded with #topics = 15.

Table D3A: Ham Data Set: Coherence Scores for LDA Models

Source: Author's creation

Number of Topics	Coherence Score
2	0.45
3	0.47
4	0.52
5	0.56
6	0.53
7	0.58
8	0.57
9	0.65
10	0.58
11	0.59
12	0.54
13	0.59
14	0.62
15	0.63
16	0.59
17	0.61
18	0.55
19	0.62
20	0.58
25	0.67
50	0.59

Figure D3A: Ham Data Set: Coherence Scores for LDA Models

Source: Author's creation

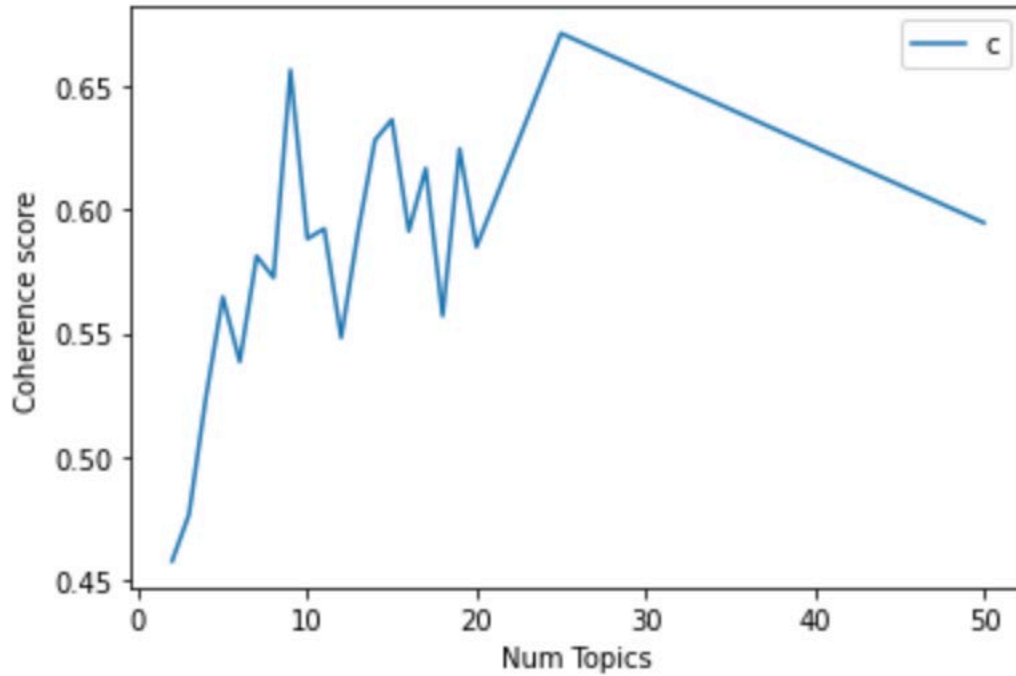


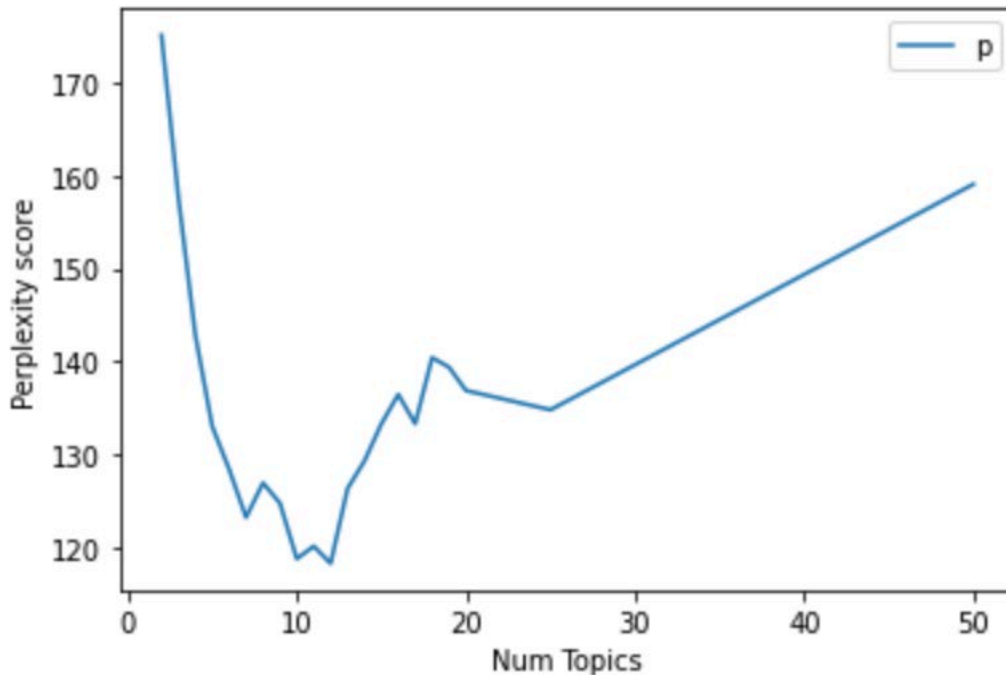
Table D3B: Ham Data Set: Perplexity Scores for LDA Models

Source: Author's creation

Number of Topics	Perplexity Score
2	175.16
3	157.68
4	142.86
5	133.01
6	128.39
7	123.21
8	126.93
9	124.76
10	118.75
11	120.11
12	118.25
13	126.32
14	129.29
15	133.29
16	136.46
17	133.30
18	140.42
19	139.38
20	136.88
25	134.80
50	159.10

Figure D3B: Ham Data Set: Perplexity Scores for LDA Models

Source: Author's creation

**Figure D3C: Ham Data Set: LDA With #Topics = 15**

Source: Author's creation

Topic 0: order customerservice shoe get size return received shoes carnival women
 Topic 1: order shoe carnival date wrote shoecarnival package narvar womens customerservice
 Topic 2: order customerservice wrote image shoe carnival women company recent information
 Topic 3: email contact office unsubscribe address new info street mailto call
 Topic 4: important font text size width max none table inherit span
 Topic 5: order shoes received delivered package pair number ordered iphone hello
 Topic 6: email order account customer phone name number contact points shoe
 Topic 7: order shoes received number pair ordered size iphone hello email
 Topic 8: business new time website email best company one year get
 Topic 9: order email shipping check details number hello item shoe customerservice
 Topic 10: shoecarnival shoe carnival wrote text women get stop men signing
 Topic 11: email message information office intended mail recipient personal return contact
 Topic 12: shoe shoecarnival carnival store text women online men get purchase
 Topic 13: font important sans text georgia open decoration none customerservice screen
 Topic 14: order shoes received pair return number ordered get size iphone

Ham Data Set: KeyBERT + LDA

For the 142,277 “cleaned and lemmatized” text of ham data, I ran KeyBERT to identify the top unigram or the top bigram storing the results in a new column. I set #topics = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 50]. I append the coherence scores and perplexity scores table and graph next.

Table D4A: Ham Data Set: Coherence Scores for KeyBERT + LDA Models

Source: Author's creation

Number of Topics	Coherence Score
2	0.58
3	0.58
4	0.60
5	0.61
6	0.61
7	0.61
8	0.60
9	0.63
10	0.61
11	0.62
12	0.62
13	0.63
14	0.64
15	0.63
16	0.64
17	0.62
18	0.64
19	0.64
20	0.65
25	0.66
50	0.57

Figure D4A: Ham Data Set: Coherence Scores for KeyBERT + LDA Models

Source: Author's creation

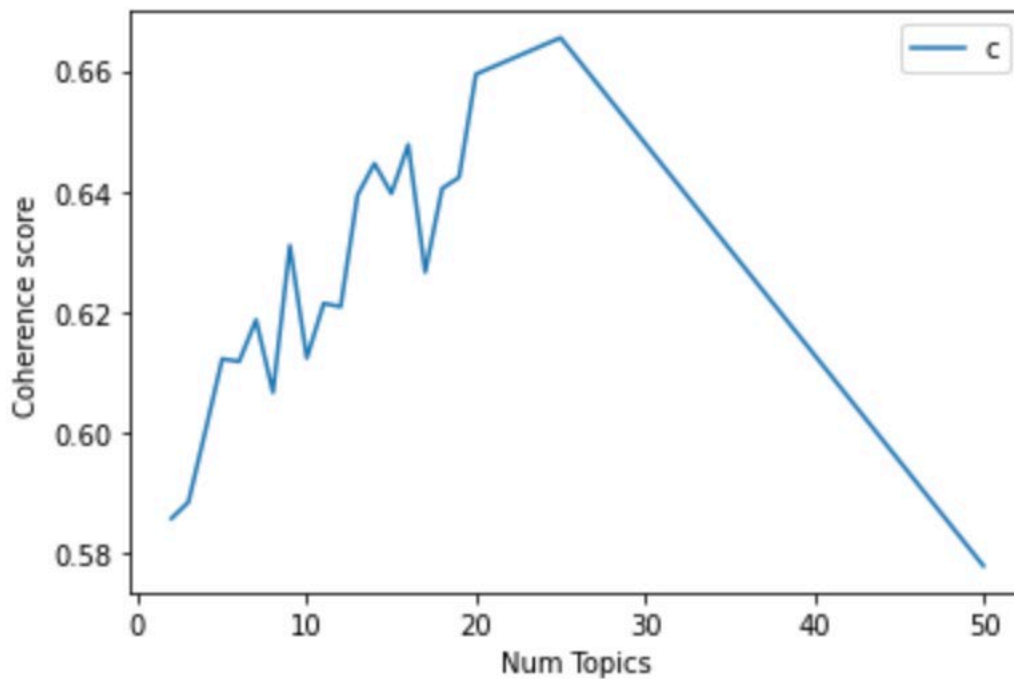


Table D4B: Ham Data Set: Perplexity Scores for KeyBERT + LDA Models

Source: Author's creation

Number of Topics	Perplexity Score
2	45.55
3	45.49
4	45.42
5	45.41
6	47.23
7	47.04
8	44.20
9	45.16
10	44.07
11	43.94
12	45.11
13	44.37
14	46.62
15	45.19
16	46.05
17	48.25
18	48.72
19	52.70
20	53.36
25	55.64
50	863521.74

Figure D4B: Ham Data Set: Perplexity Scores for KeyBERT + LDA Models

Source: Author's creation

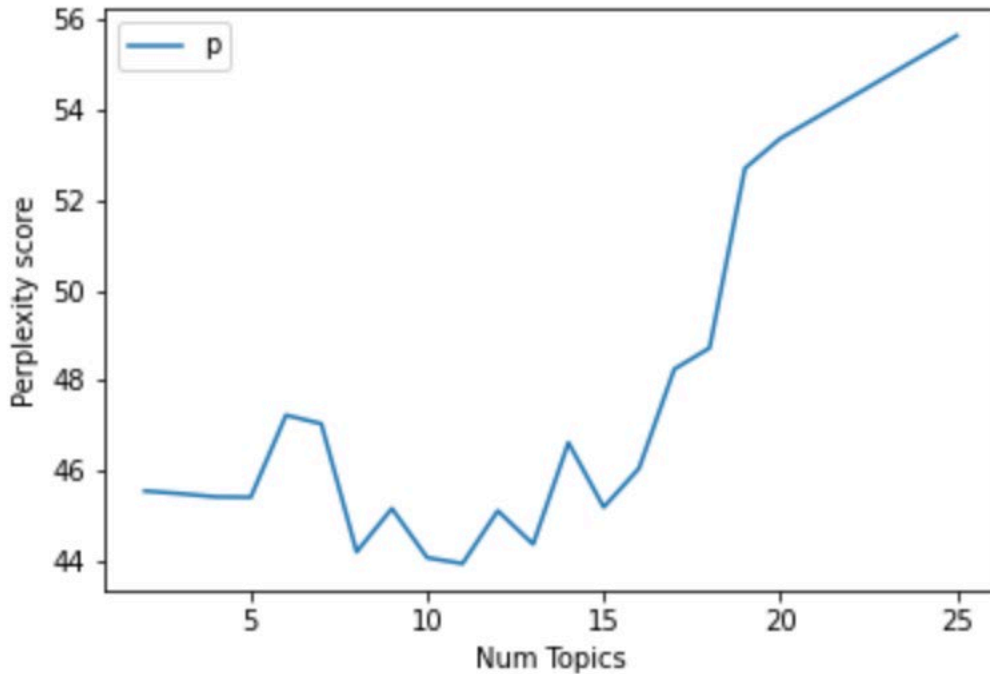


Figure D4C: Ham Data Set: KeyBERT + LDA With #Topics = 14

Source: Author's creation

- Topic 0: shipping refund order shoes response expect free customer return delivery
- Topic 1: shoe perks customerservice account carnival points chat perk rewards reward
- Topic 2: order shoes received delivered iphone refund arrived package receive cancel
- Topic 3: shoes return ordered exchange size font received boots pair crocs
- Topic 4: address email password wrong shipping charged card reset change order
- Topic 5: shoes ordered received boots pair pairs size purchased crocs sneakers
- Topic 6: order refund shoes received cancel moved cancelled donor canceled card
- Topic 7: order cancel number date cancelled tracking delivery canceled status details
- Topic 8: shoecarnival shoe carnival wrote customerservice perks message mail messages text
- Topic 9: email order emails chat unsubscribe confirmation stop remove iphone unsubscribed
- Topic 10: order cancel coupon cancelled code account orders payment purchase rewards
- Topic 11: fedex tracking return office email contact order refund iphone shipping
- Topic 12: shoes ordered order size shoe email boots refund sneakers received
- Topic 13: shoes delivered package received order arrive fedex arrived tracking iphone

Ham: KeyBERT + Guided LDA

At this stage, I had key unigram or key bigram for each of 148,103 ham emails. Next, I manually read each key unigram/bigram and manually grouped them into three topics.

Table D5A: Ham Data Set: Results from KeyBERT

Topic	Seed Words
-------	------------

Order	'order', 'status', 'tracking', 'zip', 'tracking', 'wrong', 'change', 'online', 'cancel', 'missing', 'item', 'shoes', 'size', 'shoe', 'boots', 'pair', 'sandals', 'exchange', 'shipping', 'tracking', 'delivered', 'shipment', 'shipped', 'pending'
Email	'email', 'unsubscribe', 'reset', 'password', 'account'
Payment	'billing', 'bill', 'visa', 'address', 'payment', 'paypal', 'points', 'purchase', 'refund', 'processed', 'klarna', 'apply', 'coupon', 'certificate', 'perk', 'redeem', 'gift', 'cards', 'discount', 'rewards', 'balance'

Figure D5B: Ham Data Set: Guided LDA with Seed Words

Source: Author's creation

Topic 0: order shoes cancel received cancelled tracking refund delivered number size
 Topic 1: shoe shoecarnival shoes carnival ordered email customerservice wrote perks password
 Topic 2: shoes refund order shipping received return address ordered card charged

Web Appendix E: Metrics for Six Models, Trained Using Three Different Data Sets

Table E1: Eight Accuracy Rates for the Six Models Trained on *Only Internal Data Set*, *Only External Data Set*, and the *Combined Data Set*

Note: If the criterion is to obtain the lowest false positive rate, the random forest trained on a combination of internal and external data set is the most accurate (4.48% false positive rate). However, if balanced accuracy is the criterion, the SVM trained on exclusively internal data set is the most accurate (87% balanced accuracy). The values are marked in bold typeface. Source: Author's creation

ML Classifier	False Positives	False Negatives	True Positives	True Negatives	Accuracy	Balanced Accuracy	F1 Score	AUC
<i>Logistic Regression</i>								
Trained on <i>only the internal data set</i>	21,212 15%	535 12%	4,091 88%	118,265 85%	0.84	0.86	0.27	0.87
Trained on <i>only the external data set</i>	70,372 49%	1,975 34%	3,851 66%	71,905 51%	0.51	0.58	0.09	0.58
Trained on the <i>combined data set</i>	23,648 17%	998 22%	3,628 78%	115,829 83%	0.83	0.81	0.22	0.81
<i>Random Forest</i>								
Trained on <i>only the internal data set</i>	9,949 7%	1,820 39%	2,806 61%	129,528 93%	0.91	0.76	0.30	0.77
Trained on <i>only the external data set</i>	15,999 11%	4,346 75%	1,480 25%	126,278 89%	0.86	0.57	0.13	0.57
Trained on the <i>combined data set</i>	6,245 4%	3,018 65%	1,608 35%	133,232 96%	0.93	0.65	0.25	0.65
<i>SVM</i>								
Trained on <i>only the internal data set</i>	18,807 13%	606 13%	4,020 87%	120,670 87%	0.86	0.87	0.31	0.87
Trained on <i>only the external data set</i>	49,355 35%	2,253 39%	3,573 61%	92,922 65%	0.65	0.62	0.11	0.63
Trained on the <i>combined data set</i>	19,348 14%	839 18%	3,787 82%	120,129 86%	0.86	0.84	0.27	0.84
<i>Voting Classifier</i>								
Using the models trained on <i>only the</i>	16,340 12%	853 18%	3,773 82%	123,137 88%	0.88	0.85	0.30	0.85

<i>internal data set</i>								
Using the models trained on <i>only the external data set</i>	46,869 33%	2,480 43%	3,346 57%	95,408 67%	0.67	0.63	0.12	0.62
Using the models trained on <i>the combined data set</i>	17,009 12%	1,170 25%	3,456 75%	122,468 88%	0.87	0.81	0.27	0.81
<i>XLNet</i>								
Trained on <i>only the internal data set</i>	180 6%	119 10%	1035 90%	2,832 94%	.93	.92	.87	.90
Trained on <i>only the external data set</i>	10 1%	17 2.96%	558 97%	1448 99%	.99	.98	.97	.98
Trained on <i>the combined data set</i>	234 5%	136 8%	1627 92%	4692 95%	.94	.93	.90	.94
<i>RoBERTa</i>								
Trained on <i>only the internal data set</i>	160 5%	131 11%	1023 89%	2,852 95%	.93	.91	.87	.92
Trained on <i>only the external data set</i>	7 1%	17 3%	558 97%	1451 99%	.99	.98	.98	.99
Trained on <i>the combined data set</i>	223 4%	139 8%	1,624 92%	4,703 95%	.94	.93	.90	.94

Figure E1: False Negative Rate for Different Models Trained on *Only Internal Data Set*, *Only External Data Set*, and the *Combined Data Set*

Source: Author's creation

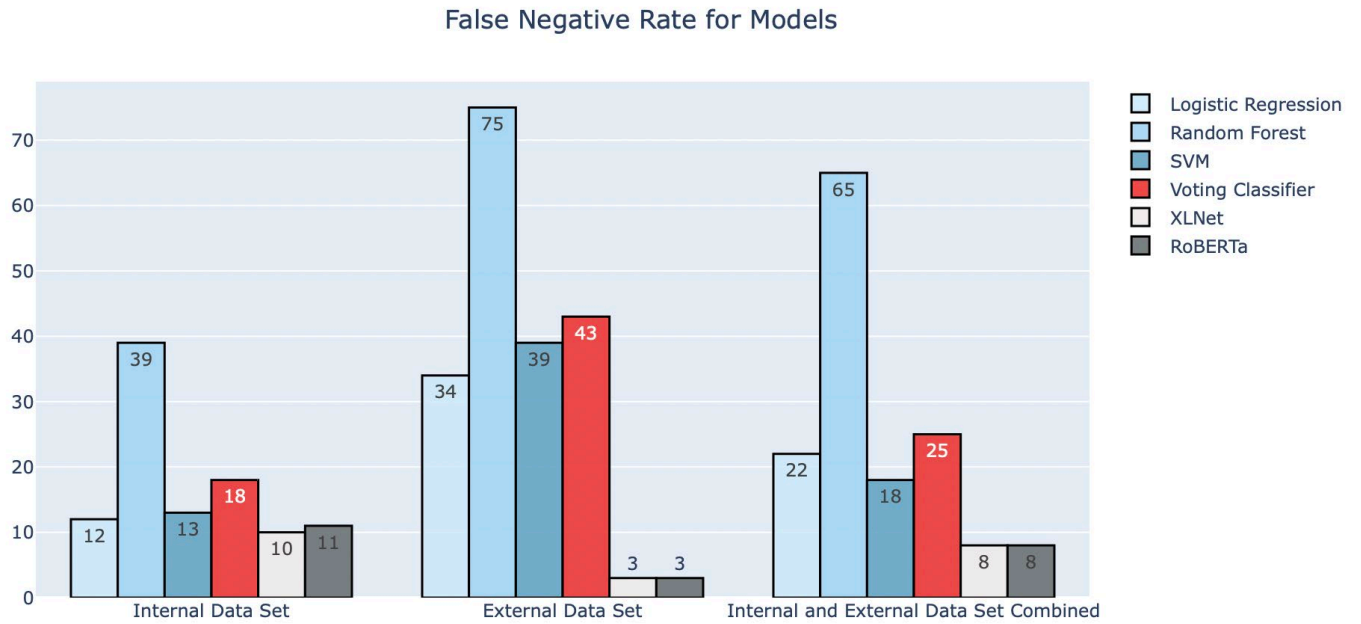


Figure E2: True Positive Rate for Different Models Trained on Only *Internal* Data Set, Only *External* Data Set, and the *Combined* Data Set

Source: Author's creation

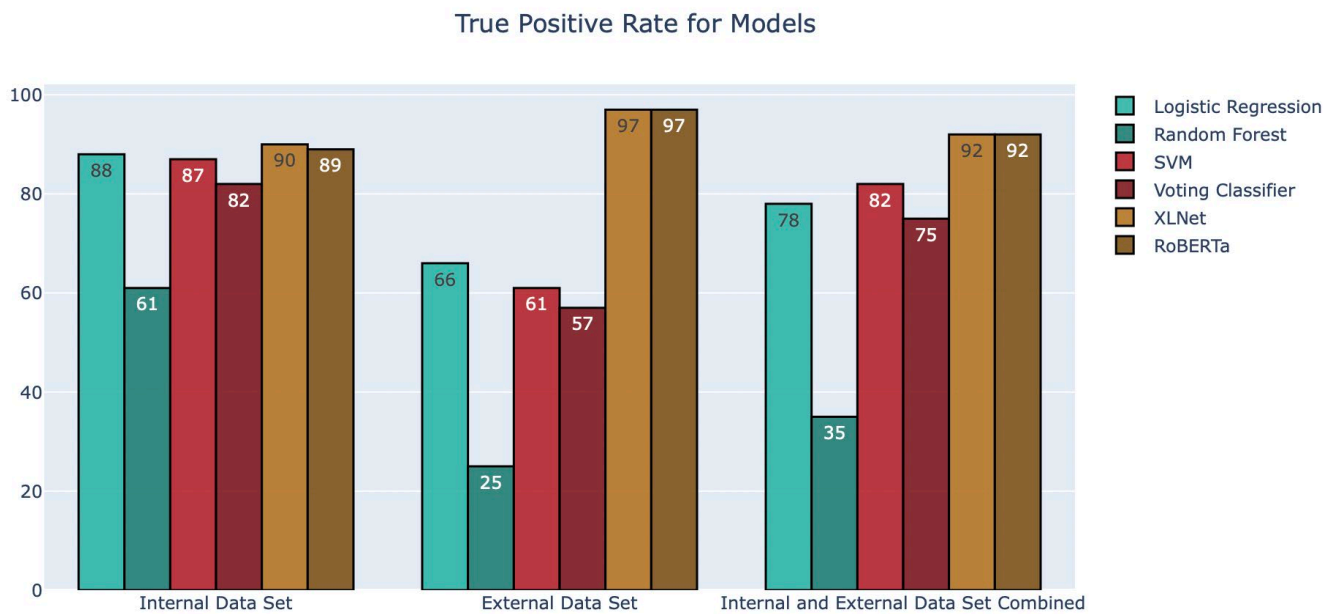


Figure E3: True Negative Rate for Different Models Trained on Only *Internal* Data Set, Only *External* Data Set, and the *Combined* Data Set

Source: Author's creation

True Negative Rate for Models

