Classifying Incoming Customer Messages for an E-Commerce Site using Supervised Learning

Misael Andrey Albañil Sánchez[0000−0003−2746−1299]
and Ixent Galpin[0000−0001−7020−6328]

Facultad de Ciencias Naturales e Ingeniería
Universidad de Bogotá – Jorge Tadeo Lozano
Bogotá, Colombia
{misaela.albanils,ixent}@utadeo.edu.co

Abstract. Throughout the world, the provision of online goods and services has increased significantly over the last few years. We consider the case of Tango Discos, a small company in Colombia that sells entertainment products through an e-commerce website and receives customer messages through various channels, including a webform, email, Facebook and Twitter. This dataset comprises 29,970 messages collected from 2019 to 2021. Each message can be categorized as being either being a *sale*, *request* or *complaint*. In this work we evaluate different supervised classification models to automate the task of classifying the messages, viz. decision trees, Naive Bayes, linear Support Vector Machines and logistic regression. As the data set is unbalanced, the different models are evaluated in combination with various data balancing approaches to obtain the best performance. In order to maximize revenue, the management is interested in prioritizing messages that may result in potential sales. As such, the best model for deployment is one that minimizes false positives in the *sales* category, so that these are processed in a timely fashion. As such, the best performing model is found to be the Linear Support Vector Machine using the Random Over Sampler balancing technique. This model is deployed in the cloud and exposed using a RESTful interface.

Keywords: E-commerce · Message classification · Supervised learning · Support Vector Machine · Balancing techniques

1 Introduccion

In Colombia there are approximately 307,679 micro-enterprises registered [5] that offer their products and services on the Internet. During 2020, according to the BlackSip report [3] online sales increased 113% over the previous year in this business sector. Although the volume of online transactions and sales in Colombia has been increasing year-on-year, 2020 represented the highest growth over last five preceding years (see Fig. 1). This trend has been further punctuated during the COVID-19 pandemic, during which many goods and services were made available online.

Fig. 1: Growth of online retail of some Latin American countries 2017–2021 [3].

For this work we consider Tango Discos[1], a small company in Colombia, as a case study. This company has over 20 year's experience in the sale of entertainment products such as music discs, books and accessories for audio equipment. In 2019, they decided to open their online store to the public and offer their catalog of products through an e-commerce website. Subsequently, the ubiquity of social networks led to the opening of new channels of communication with customers. This, in turn, led to a substantial increase in incoming message traffic with requests, complaints or questions about the products. The high volume of requests has proved challenging due to the finite capacity of the customer service department. An important concern for the management was knowing that a significant percentage of incoming messages had the purpose of requesting information about products. Such interactions have a high probability of resulting in a subsequent sale. Due to inefficiency in the handling of incoming messages, there is concern that potential sales are being lost.

In this paper, using a dataset of incoming messages in Spanish from various sources provided by the company, we evaluate different approaches to classifying the messages in an automated fashion. Each free-text message in the data set is labelled to indicate whether it is a *sale*, *request*, or *complaint*. Classification models are trained using supervised techniques such as decision trees, Naive Bayes, linear Support Vector Machines (LSVMs) and logistic regression. These are couple with various techniques used to mitigate the fact that the data is unbalanced. Using confusion matrices, Receiver Operating Characteristic (ROC) curves, and metrics such as accuracy and recall, the best performing models are identified. The best performing model is subsequently deployed exposing a RESTful interface on a Heroku server, available for the business to use so that it may prioritize messages according to its requirements.

In this paper, we broadly follow the well-established CRISP-DM methodology for data mining projects [16], and this is reflected in the paper structure. The *business understanding* phase is covered in this section, as well as Section 2, which presents related works with a brief survey of message classification ap-

[1] `https://tangodiscos.com.co/`

proaches used in various domains. The *data understanding* and *data preparation* phases are described in Sections 3 and 4. Section 5 corresponds to the *modelling* phase, and presents the models and balancing techniques implemented, which are evaluated subsequently in Section 6 (*evaluation* phase). The *deployment* phase is presented in Section 7. Finally, Section 8 draws conclusions and proposes future work.

2 Related Work

One of the earliest works that address the issue of message classification is proposed by Busemann *et al.* [4], who present an approach to classify customer emails using various approaches including Naive Bayes and Support Vector Machines. The aim of the approach is to identify the client's issue thus partially automating the technical support process.

Considerable work in message classification addresses the issue of detecting spam emails. An early proposal is that of Duan [6] *et al.*, who describe a binary classification model to determine whether an email is spam, using a KNN classification algorithm. More recently, Alghoul *et al.* [2] address the issue of spam email classification using an artificial neural network. A comprehensive survey of machine learning approaches in this domain is given by Mansoor *et al.* [9].

The widespread use of social networks like a Twitter, Facebook and Instagram by teenagers has led to the phenomenon of *cyberbullying*. Menini [11] *et al.* propose a natural language processing (NLP) approach to allow the detection of different forms of abusive language by means of a classification model. Cyberbullying is also the focus of other proposals [17,14].

In the e-commerce domain, which is the focus of our paper, the role of message classification has also been shown to be invaluable [13,10,1]. For example, work by Nkansah underlines how in Ghana, e-commerce is becoming an important sales channel for many companies. The work proposes an approach whereby users interact with other users and products. The result is a convenient, intuitive and contextually relevant e-shopping system that, based on user messages, identifies categories of products. This is complemented with a recommendation algorithm to provide customers with a personalized list of product recommendations available in the online store [13].

3 Integration of Data Sources

The dataset used in our case study contains 29,970 records (incoming messages) from different sources over a period of three years (March 2019 to December 2021). For each source, scripts were developed using different technologies to automate the extraction and loading of messages and corresponding metadata into a MySql[2] database, summarized in Figure 2. The messages are loaded to the database from each source as follows:

[2] `https://www.mysql.com/`

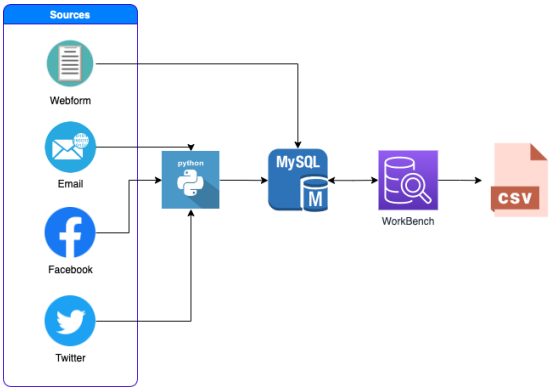


Fig. 2: Information sources Diagram

- **Webform** The form on the e-commerce website[3]. This form is developed in PHP and interacts directly with the databas using SQL scripts.
- **Email** The company has a corporate email available to the public. A script was developed in Python 3.9 using version 0.1.2 of the imaplib library[4], and version 0.1.0 of the email[5] library, to extract the messages and send them to the database.
- **Facebook** The account on this social network has a chat for users to send messages. A script was developed in Python and working with Facebook Messenger Graph API[6] the messages are captured and sent it to the database.
- **Twitter** The account on this social network allows the company to receive tweets or mentions from their users. A script was developed in Python using version 1.7.1 of the tweepy[7] library to capture the tweets and send then to the database.

Table 1: Number of messages per source between 2019 to 2021.

Source	Percentage of messages	Message count
Webform	64	19,181
Email	15	4,496
Facebook	13	3,896
Twitter	8	2,398
Total	100	29,970

[3] https://tangodiscos.com.co/webform/contacto
[4] https://pypi.org/project/secure-imaplib/
[5] https://pypi.org/project/email/
[6] https://developers.facebook.com/docs/graph-api
[7] https://pypi.org/project/tweepy

The figures presented in Table 1 show the number of messages received during the three-year period from the different sources. Once the message records were loaded to the database, the open source Workbench[8] software was used to connect to the MySql database and export the records to CSV format for further processing. The message records obtained have the following attributes:

- `Id`: Primary key generated automatically by the database;
- `Source`: Identifier of the message source;
- `Submitted Time`: Date and time the message was sent;
- `IP address`: IP address of the device from which the message is sent;
- `Name and Last name`: Name and surname of the sender of the message;
- `Email`: Email Contact of the sender;
- `Phone`: Telephone number of the sender;
- `Comment`: Message text;
- `Type`: Classification or label manually assigned by the customer service agent (either *sale*, *request*, *complaint* or *spam*).

4 Data Preparation

As an initial step, messages from the Trash category are filtered out, leading to the removal of 4,140 spam messages or those that had irrelevant information. The resulting data set has 25,830 messages, manually classified on three remaining labels (*Sale*, *Request* and *Complaint*). Table 2 shows the count of messages classified by each of those labels.

Table 2: Number of messages per label.

Labels	Messages Count
Sales	15,400
Request	9,020
Complaints	1,410
Total	25,830

In this study, the message text is the focus of analysis. To extract the relevant information of each message, we apply the following steps which are well-established in text-mining [8]:

1. *Tokenization* [15], whereby the text is split into words. For this, we use the Python spaCy[9] library.
2. *Cleaning*, whereby elements that do not contribute semantics of the message such as emoticons, hashtags, URLs etc. are removed.

[8] `https://www.mysql.com/products/workbench/`
[9] `https://spacy.io`

3. *Stop-word removal*, where words that do not provide much information are removed. Examples of such words include conjunctions or prepositions. The spaCy library provides us a dictionary of stop-words[10] in Spanish with a list of these words that can be filtered.
4. *Normalization* of the text [8], whereby the tokenizer recognizes words equivalent words, e.g., by ignoring capitalization (thus 'buy', 'Buy' y 'BUY' would be considered the same word).
5. *Lemmatization* [8], a process which obtains the root of a word according to its lexical component. For example, "buys", "would buy", "bought" would all map to to "buy". By applying lemmatization, it is possible to reduce the number of unique words.

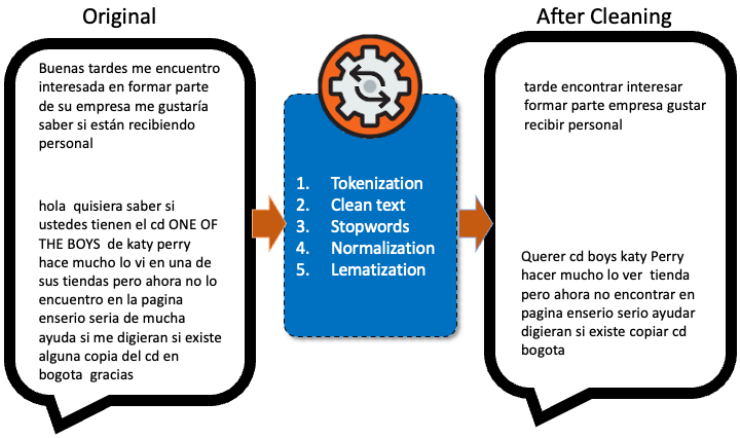


Fig. 3: Example of messages before and after the preparation workflow

Figure 3 presents an example of applying these five steps. After cleaning, the tokenized elements within a new list are shown in Figure 4. Next, we proceed to the generation of the TF-IDF matrix. The TF-IDF matrix [8] is a technique used to quantify words within a document, this value is calculated for each word and represents the relevance in the document and overall corpus.

$$tfidf(t, d, D) = tf(t, d)idf(t, D) \tag{1}$$

Figure 5 presents a fragment of the obtained values of the TF-IDF matrix. Each value represents the weight of the word not only with respect to the vocabulary index and also considering all documents.

As previously noted in Table 2, the data is clearly unbalanced. If the classification methods are applied directly to the data set, the model is likely to favor

[10] `https://spacy.io/models/es`

SALES	PETITIONS	COMPLAINTS
querer	preguntar	mirar
comprar	realizar	desconfiar
tener	entregar	atrazar
pedir	enviar	necesitar
informar	estudiante	consignar
album	manifestar	pasar
disco	formar	esperar
producto	ayudar	concertar
interesar	aclarar	hacer

Fig. 4: Example snippet list of words normalized by label.

	Sales	Petitions	Complaints
840	0,157885	0,154245	0,169823
841	0,167443	0,123553	0,393455
842	0,112398	0,02112	0,235832
843	0,331745	0,154231	0,152712
844	0,124234	0,238463	0,136847
845	0,245456	0,275629	0,08624
846	0,331745	0,188452	0,241839
847	0,163829	0,473056	0,246728
848	0,230584	0,148234	0,108342

Fig. 5: Snippet matrix TF-IDF values.

the label with the greatest number of observations. To avoid this bias, we apply one of the following balancing techniques (at a time) over the dataset:

- *Ramdom Oversampler* [12], which involves randomly duplicating instances of the minority class and adding them to the training dataset.
- *Random Undersampler* [12], which conversely has the effect of reducing the number of instances in the majority class.
- *Smote*, a synthetic minority oversampling technique which analyzes minority samples and artificially synthesizes new samples based on minority samples, and adds them to the data set.
- *Nearmiss* [7], a undersampling technique whose objective is based on reducing the number of samples with the majority label. Its choice for the elimination of the instances employs the k-nearest neighbors approach .
- *Weight*, works as an equalizer between the different labels so that the classification model is as equitable as possible between them. For the present study, we use the Sci-kit Learn library[11] to calculate each label weight.

[11] `https://scikit-learn.org/stable/index.html`

5 Modeling

In this section, we evaluate combinations of the balancing techniques described in the previous section with various models, namely:

- *Decision Tree Classifier*, one of the predictive modeling approaches that uses a decision tree to work on the observations on an item (represented in the branches) and obtain predictions about the target value of the item (represented in the leaves).
- *Naive Bayes*, a probabilistic machine learning model based on Bayes' theorem, where we can find the probability that A occurs, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is, the presence of one particular feature doesn't affect the other. That is why it is referred to as naive.
- *Linear Support Vector Machine*, an algorithm that determines the best decision between vectors that belong to the same group (or category) and the vectors that do not belong to the same category. It can be applied to any type of vectors encoding any type of data. To take advantage of this classification method, messages must to be transformed into vectors.
- *Logistic Regression*, a type of statistical model is often used for classification and predictive analysis where the probability of an event occurring is estimated based on a given data set of independent variables. Since the result is a probability, the dependent variable is limited between 0 and 1. In logistic regression, a "logit" transformation is applied over the probability, that is, the probability of success divided by the probability of failure.

		Classifier Model			
		Decision Tree Classifier	Bayes	Linear Support Vector Machine	Logistic Regression
Balancing Method	Unbalanced	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]
	Over Sampler	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]
	smote	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]
	Random Under Sampler	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]
	NearMiss	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]
	Weight	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]	[ROC Curve][Confusion Matrix][Acurracy][Recall]

Fig. 6: Combinations of cases to be analyzed.

Figure 6 presents the combinations of classification model and balancing method evaluated. The unbalanced method refers to the case when no balancing

method is applied to the data. For each combination in our search space, the following metrics are computed to evalute its performance:

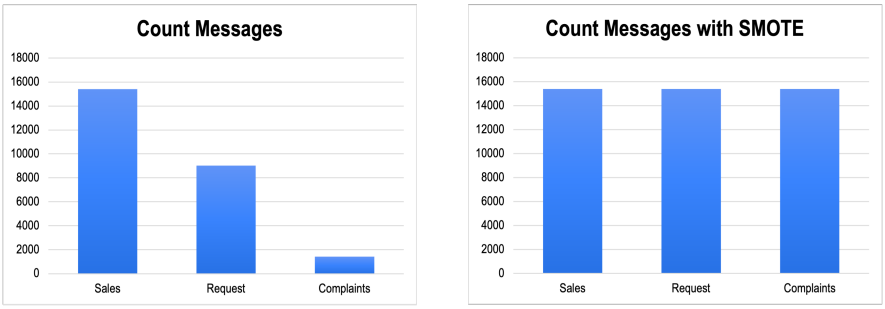
- *ROC Curve*, the representation of the proportion of true positives (VPR = True Positives) versus the proportion of false positives (FPR = False Positives). Interpretation is based on comparison of the area under the curve (AUC) of the tests. This area has a value between 0.5 and 1, where 1 represents a perfect diagnostic value and 0.5 is a test without diagnostic discriminatory capacity.
- *Confusion matrix*, whereby each column of the matrix represents the number of predictions of each class, and each row represents the instances in the actual class.
- *Precision*, a metric which enables the quality of the machine learning model in classification process to be measured. In this case of study, it refers to the fact that precision is the answer to the question: ¿What percentage of messages were True Positives (VPR)?
- *Recall*, which calculates how many of the Actual Positives the model capture through labeling it as Positive (True Positive). Applying the same understanding, we know that Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negatives.
- *F-Value*, or *F1-value* is used to combine the precision and recall measures into a single value, using the harmonic mean of these two values.
- *Accuracy*, measures the fraction of cases which are correct.

For the development of the analysis in this study, we use the Anaconda Navigator 2.1.4, Spyder 5.1.5 and Python 3.9.0. To perform the data balancing, we use the imbalanced-learn[12] library, which possess methods corresponding to each balancing technique in Figure 6. The classification models are implemented using the sklearn[13] library. In addition, we use the numpy and pandas libraries for data extraction and analysis, as well as matplotlib and scikitplot for result visualization. The Python code and anonymized version of the data set we use is available on GitHub[14]. As an illustrated example, we explain the steps carried out for the DecisionTreeClassifier and the data set balanced with the smote technique, one of the combinations in the Figure 6:

- The data set is obtained from the source file in CSV format, as shown in Figure 2). Subsequently, data set is divided into training and test partitions using the train-test-split[15] library.
- After splitting the dataset, the next step involves feature engineering. The text messages will be converted into a count matrix of tokens (CountVectorizer), and then will be transformed into a count matrix in a normalized TF-IDF representation (TF-IDF transformer).

[12] https://imbalanced-learn.org/stable/index.html
[13] https://scikit-learn.org/stable/modules/classes.html
[14] https://github.com/malbanil/MIAD-Classifying-Messages.git
[15] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

– The Smote balancing technique is implemented to obtain a balanced data set (see the result in Figure 7).
– We use a function with the algorithm that implements the classification model on the balanced data set. This function is parameterized by balancing method.
– The classification model performance metrics are obtained and stored for later analysis (see precision, recall, F1-score in Figure 8, and the ROC curve and confusion matrix in Figure 9).



(a) Unbalanced data set. (b) Balanced data set with Smote.

Fig. 7: Data set comparison before and after applying the Smote balancing technique.

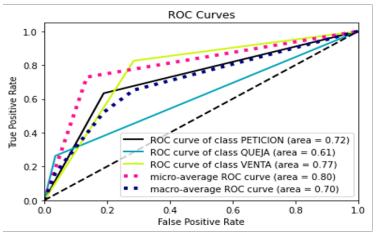
accuracy 0.6503225806451612

	precision	recall	f1-score	support
Sales	0.58	0.49	0.53	278
Petitions	0.16	0.29	0.20	38
Complaints	0.76	0.78	0.77	459
Accuracy			0.65	775
Macro avg	0.50	0.52	0.50	775
Weighted avg	0.67	0.65	0.66	775

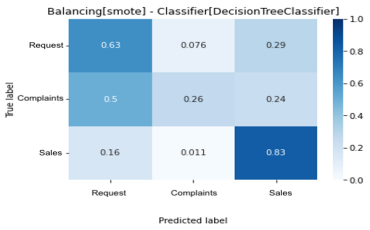
Fig. 8: Results of the classification model over the test data.

6 Evaluation

This section reports the results obtained for the combinations of classification model and balancing method in Figure 6). The accuracy and recall for each combination are shown in Figure 10. In Figures 11 and 12, confusion matrices and ROC curves are shown respectively. Note that due to space constraints, we do not show the results for all combinations, only the ones that we deem to be most “interesting” or representative.



(a) ROC Curve.



(b) Confusion matrix.

Fig. 9: Graphs resulting from the process using the DecisionTreeClassifier classification model and the balanced data set with Smote.

We observe that if the results are only evaluated by taking the accuracy criterion into account, the "best" combination would be the Linear Support Vector Machine with no data balancing, as the accuracy value is the highest at 0.7896. However, taking into account the confusion matrix in the analysis (see Figure 11(c)), we observe that this combination underperforms when classifying "Complaint" messages. Therefore, business requirements may deem it to be inadequate for deployment.

If we take as reference the confusion matrices (see Figure 11(a)) and ROC curves (see Figure 12(a)), we may conclude that the combination that best classifies the three labels is Naive Bayes with smote data balancing. This is reflected by the fact that the values of the confusion matrix diagonal are the highest and most balanced. Thus, if all three labels are deemed to be equally important according to business requirements, this would be the best combination for deployment.

However, if we consider the concern exposed in the business understanding phase, where those messages that correspond to sales are more important than other types of messages, the best combination for deployment would be the Linear Support Vector Machine classification model coupled with the Random Over Sampler data balancing technique (see Fig. 11(b)).

7 Deployment

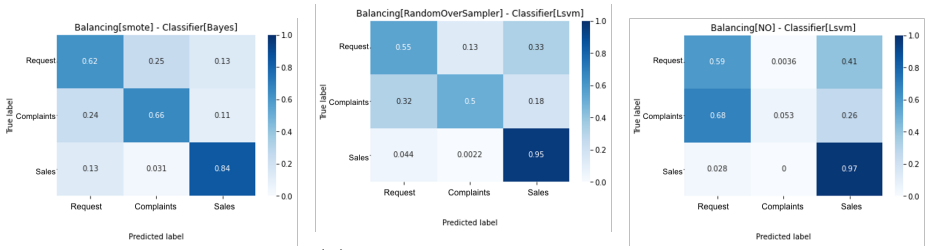
For the deployment phase, the model exposes an interface via a REST API that has two input parameters, the message content and source, and returns the predicted message class:

- Input: `Message(String), Source(String)`. Sender message and source of the message.
- Output: `Label(String)`. The classification label that the model predicts.

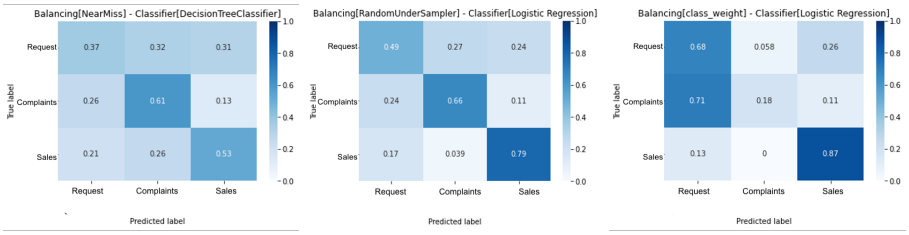
ACURRACY				
	DecisionTreeClassifier	Bayes	LSVM	Logistic Regression
Unbalanced	0,74509803922	0,78322580645	0,78967741935	0,76000000000
over_sampler	0,66322580645	0,74322580645	0,78580645161	0,75870967742
smote	0,71354838710	0,75483870968	0,77290322581	0,76903225806
RandomUnderSampler	0,51225806452	0,62193548387	0,66451612903	0,67354838710
NearMiss	0,44516129032	0,68387096774	0,62322580645	0,64645161290
Weight	0,69161290323		0,78193548387	0,76903225806

RECALL				
	DecisionTreeClassifier	Bayes	LSVM	Logistic Regression
Unbalanced	0,51	0,52	0,54	0,57
over_sampler	0,52	0,69	0,67	0,57
smote	0,57	0,71	0,62	0,69
RandomUnderSampler	0,52	0,69	0,64	0,64
NearMiss	0,5	0,57	0,58	0,58
Weight	0,58		0,62	0,58

Fig. 10: Accuracy and recall values.



(a) Bayes - Smote (b) LSVM - RandomOver-Sampler (c) LSVM - Unbalanced



(d) DecisionTreeClassifier - Nearmiss (e) Logistic Regression - RandomUnderSampler (f) Logistic Regression - Weight

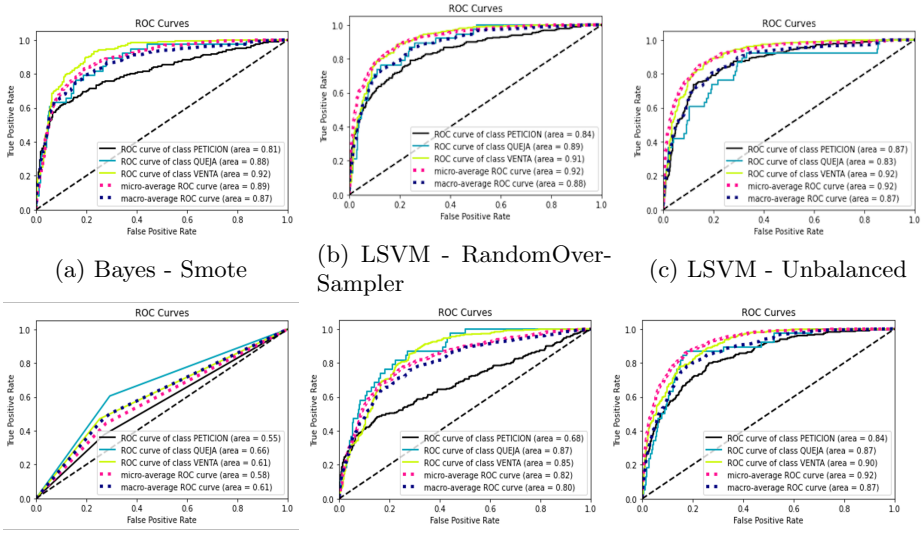
Fig. 11: Confusion matrix results graphs.

This solution is hosted in a Heroku cloud server instance, back-end framework with Python 3.1.9 and Flask 2.1[16] library to create a RESTful services, as shown in the implementation architecture in Figure 13.

8 Conclusions and Future Work

As this case study demonstrates, varying classification models alone, in some scenarios, is insufficient to obtain a good result. The data preparation and bal-

[16] `https://flask.palletsprojects.com/en/2.1.x/`



(a) Bayes - Smote

(b) LSVM - RandomOverSampler

(c) LSVM - Unbalanced

(d) DecisionTreeClassifier - Nearmiss

(e) Logistic Regression - RandomUnderSampler

(f) Logistic Regression - Weight

Fig. 12: ROC curve results graphs.

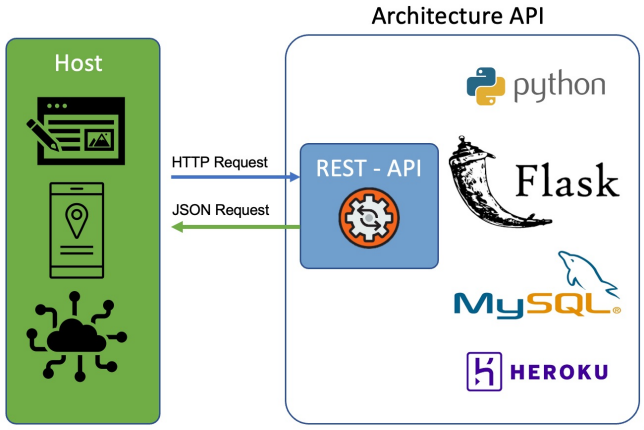


Fig. 13: Deployment implementation architecture.

ancing techniques played an important part in improving model performance. Furthermore, in addition to the technical process, it was necessary to evaluate the results of applying different classification models in combination with balancing techniques. With this information, concepts from data analytics together with the business objectives were used to determine the best model.

This work paves the way for future studies that seek to understand how e-commerce clients communicate with businesses to make requests, complaints or ask about products (which potentially result in sales). It is envisaged that this can facilitate the development of tools that enhance help desks or customer management services. Future work could usefully include: (1) Evaluation of the selected model with a naturally balanced data set; (2) The evaluation of other classification methods such as neural networks, genetic algorithms to compare their performance with the results obtained; (3) Incorporation of a sentiment analysis algorithm to improve classification accuracy; (4) Integrating message classification with a recommender system to give website visitors personalized product recommendations.

References

1. Adaji, I., Kiron, N., Vassileva, J.: Evaluating the susceptibility of e-commerce shoppers to persuasive strategies. a game-based approach. In: International Conference on Persuasive Technology. pp. 58–72. Springer (2020)
2. Alghoul, A., Al Ajrami, S., Al Jarousha, G., Harb, G., Abu-Naser, S.S.: Email classification using artificial neural network (2018)
3. BlackSip, Vtex, Nielsen, PayU, Credibanco, MercadoLibre, Rappi, emBlue, Icommkt: BlackIndex: reporte del ecommerce en Colombia. BlackSip (2019)
4. Busemann, S., Schmeier, S., Arens, R.G.: Message classification in the call center. arXiv preprint cs/0003060 (2000)
5. Confecamaras: `https://confecamaras.org.co` (13 de Enero de 2022)
6. Duan, L., Li, A., Huang, L.: A new spam short message classification. In: 2009 First International Workshop on Education Technology and Computer Science. vol. 2, pp. 168–171. IEEE (2009)
7. Fang, W., Luo, H., Xu, S., Love, P.E., Lu, Z., Ye, C.: Automated text classification of near-misses from safety reports: An improved deep learning approach. Advanced Engineering Informatics **44**, 101060 (2020)
8. Manning, C., Raghavan, P., Schütze, H.: Introduction to information retrieval. Natural Language Engineering **16**(1), 100–103 (2010)
9. Mansoor, R., Jayasinghe, N.D., Muslam, M.M.A.: A comprehensive review on email spam classification using machine learning algorithms. In: 2021 International Conference on Information Networking (ICOIN). pp. 327–332. IEEE (2021)
10. Masterov, D.V., Mayer, U.F., Tadelis, S.: Canary in the e-commerce coal mine: Detecting and predicting poor experiences using buyer-to-seller messages. In: Proceedings of the Sixteenth ACM Conference on Economics and Computation. pp. 81–93 (2015)
11. Menini, S., Moretti, G., Corazza, M., Cabrio, E., Tonelli, S., Villata, S.: A system to monitor cyberbullying based on message classification and social network analysis. In: Proceedings of the third workshop on abusive language online. pp. 105–110 (2019)
12. Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th international conference on information and communication systems (ICICS). pp. 243–248. IEEE (2020)
13. Nkansah, E.A.: Kayayo: An e-commerce site with recommendations and text messaging (2013)

14. Özel, S.A., Saraç, E., Akdemir, S., Aksu, H.: Detection of cyberbullying on social media messages in turkish. In: 2017 International Conference on Computer Science and Engineering (UBMK). pp. 366–370. IEEE (2017)
15. Webster, J.J., Kit, C.: Tokenization as the initial phase in nlp. In: COLING 1992 volume 4: The 14th international conference on computational linguistics (1992)
16. Wirth, R., Hipp, J.: Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. vol. 1, pp. 29–39. Manchester (2000)
17. Zois, D.S., Kapodistria, A., Yao, M., Chelmis, C.: Optimal online cyberbullying detection. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2017–2021. IEEE (2018)