Aravamuthan, Sarang, Jogalekar, Prasad, and Lee, Jonghae (2022) Extracting features from textual data in class imbalance problems. Journal of Computer-Assisted Linguistic Research 6: 42-58. ISSN: 2530-9455.

Extracting Features from Textual Data in Class Imbalance Problems

Sarang Aravamuthan*, Prasad Jogalekar, Jonghae Lee

Ericsson Inc., 2755 Augustine Dr., Santa Clara, CA 95054, USA

*Corresponding author: sarang.aravamuthan@ericsson.com

Received: 15 August 2022 / Accepted: 9 October 2022 / Published: 25 November 2022

Abstract

We address *class imbalance* problems. These are classification problems where the target variable is binary, and one class dominates over the other. A central objective in these problems is to identify features that yield models with high precision/recall values, the standard yardsticks for assessing such models. Our features are extracted from the textual data inherent in such problems. We use n-gram frequencies as features and introduce a *discrepancy score* that measures the efficacy of an n-gram in highlighting the minority class. The frequency counts of n-grams with the highest discrepancy scores are used as features to construct models with the desired metrics. According to the best practices followed by the services industry, many customer support tickets will get audited and tagged as "contract-compliant" whereas some will be tagged as "over-delivered". Based on in-field data, we use a random forest classifier and perform a randomized grid search over the model hyperparameters. The model scoring is performed using an F_β scoring function. Our objective is to minimize the follow-up costs by optimizing the recall score while maintaining a base-level precision score. The final optimized model achieves an acceptable recall score while staying above the target precision. We validate our feature selection method by comparing our model with one constructed using frequency counts of n-grams chosen randomly. We propose extensions of our feature extraction method to general classification (binary and multi-class) and regression problems. The discrepancy score is one measure of dissimilarity of distributions and other (more general) measures that we formulate could potentially yield more effective models.

Keywords: class imbalance, feature selection, n-gram frequency, NLP techniques, random forest classifier.

DOI: HTTPS://DOI.ORG/10.4995/JCLR.2022.18200
PUBLISHER: UNIVERSITAT POLITÈCNICA DE VALÈNCIA.

1. Introduction

In a supervised learning problem, the goal is to build a model to predict a variable of interest (the target) based on the values of other variables (the feature set). The model is built from training (or labeled) data for which the target values are known. The model can be a classifier or a regressor depending on whether the target variable is discrete or continuous. Several algorithms exist to build these models such as Random forest, Linear regression, or Support vector machines.

Of particular interest are *class imbalance* problems. These are classification problems where the target variable is binary, and one class (usually called the negative class) dominates over the other (positive class). In these problems, normally more than 90% of the dataset belong to the negative class. Such problems occur in diverse fields such as banking, healthcare or technology and some examples are:

- Identification of defaulters in loan applicants (most are unlikely to default).
- Classification of a tumor as benign or malignant based on patient symptoms (most are benign).
- Classification of an email as ham or spam based on its text (most are ham).
- Identification of stocks that are likely to go 10x up in a few years (most are not).

Some typical challenges posed by these problems include:

- Defining a scoring function to rate the classifier: A naïve classifier that predicts all instances as negative will automatically obtain a high accuracy since a majority of the instances are assumed to belong to the negative class. Thus, a more nuanced scoring criterion is required to measure the effectiveness of the classifier. Some typical metrics used are precision, recall and F1-score (more generally, a F_β score).
- Identifying features that accentuate the minority class: Often there are many features in the dataset present either explicitly or derived implicitly from the metadata associated with the data. The challenge then is to identify traits that highlight the minority class. Such features lead to models with improved precision/recall values.

Some common approaches to address such problems include:

- Oversampling of minority class. Here multiple copies of the minority class are selected to increase its proportion in the training set.
- Undersampling of majority class whereby several instances of the majority class are discarded in the training set.
- Synthetic minority oversampling (SMOTE) whereby new instances of the minority class

are artificially realized through interpolation methods.

- Cost-sensitive training. Here the idea is to increase the cost of classification errors on the minority class.

For a detailed exposition of these methods, see (He and Garcia, 2009, Chawla, 2010). However, each of these approaches have their drawbacks. For example, undersampling can result in missing some concepts pertaining to the majority class resulting in a poorly trained model. The problem with oversampling is more subtle: since this process just adds more copies of minority class instances to the original dataset, the model becomes too tailored to the training set leading to overfitting. In particular, the model will perform well on the training data but poorly on the test data. This is true even when the standard method of using cross-validation is used to prevent overfitting since the model has already seen some of the observations in the holdout set and memorized their labels (Santos et al. 2018). The SMOTE method also presents some challenges such as overlapping between classes and overgeneralization. For further details as well as some workarounds, such as the Borderline-SMOTE algorithm or the Adaptive Synthetic Sampling (ADA-SYN) algorithm, see (He and Garcia, 2009).

One possibility for extracting features is from textual data. There is often a significant amount of unstructured text associated with each training observation. These could be experts' opinions or updates to a problem from field engineers or simply text inherent in the data (e.g., the body of an email). Identifying any feature from this data that would emphasize the minority class would yield highly effective models.

The features that are normally extracted from text are frequencies of specific words (or, more generally, n-grams). The challenge is to identify the right n-grams whose frequency counts would yield improved models.

In the next sections, we review class imbalance learning and discuss some prior work in this area. In Section 4, we propose our method of feature extraction. In Section 5, we provide the implementation results on a dataset consisting of customer support tickets of which a small fraction are labeled over-deliveries. In Section 6, we propose extensions of our idea to other supervised learning problems. Section 7 summarizes our results and relates our work to other challenging issues with class imbalance problems such as class overlapping.

2. The Class Imbalance Problem

The standard machine learning algorithms implicitly assume a uniform distribution of the target among the two classes in the training data. When this assumption fails, then the algorithm when applied normally will automatically favor the majority class.

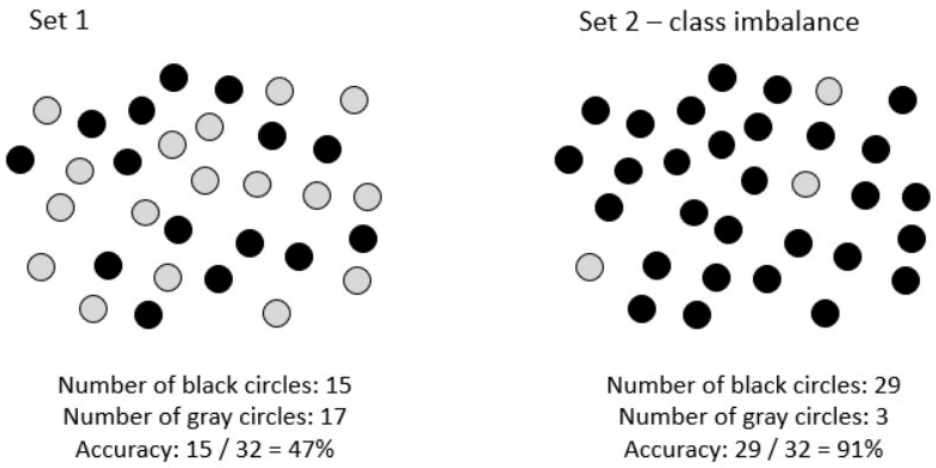
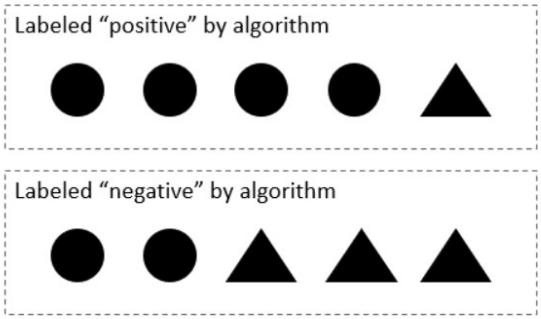


FIGURE 1. AN ILLUSTRATION OF NAÏVE CLASSIFICATION ON UNIFORM DISTRIBUTION AND CLASS IMBALANCE.

In real-life scenarios, however, accurate prediction of minority class is often of vital importance. Examples abound such as fraud detection, anomaly prediction, tumor identification etc. Here, the standard metric of using accuracy to decide the quality of classification is no longer applicable. We illustrate this in Figure 1, where Set 1 shows a uniformly distributed set of gray and black circles, whereas Set 2 clearly has a class imbalance. Consider a classification model that labels every circle as black. The accuracy would be low for Set 1 but very high for Set 2. Although the model would accurately classify 91% of the time in Set 2, it fails at its original objective – to identify the minority class.

To counter this, alternate metrics are used to score classifiers. The two standard ones are *precision* and *recall*. Precision is the ratio of the number of true positive predictions to the number of positive predictions by the classifier. Thus, classifiers with high precision will have a lower number of misclassifications of the predicted class than as the positive class. Recall is the ratio of the number of true positive predictions by the classifier to the actual number of positives in the data. Thus, classifiers with high recall will have a high percentage of correct classifications of the positive class. A calculation of these quantities is shown in Figure 2 where circles are considered as belonging to the positive class and triangles as belonging to the negative class.



$$\text{Precision} = \frac{\text{Number of true positives}}{\text{Number of labeled positives}} = \frac{4}{5}$$

$$\text{Recall} = \frac{\text{Number of true positives}}{\text{Number of actual positives}} = \frac{4}{6}$$

$$\text{Accuracy} = \frac{\text{Number of true positives+true negatives}}{\text{Total number of data points}} = \frac{7}{10}$$

FIGURE 2. COMPUTATION OF PRECISION, RECALL AND ACCURACY

It's easy to realize classifiers with high precision or recall. A classifier that always predicts positive will have 100% recall at the cost of low precision. On the other hand, a classifier that classifies everything as negative will have zero recall. The challenge thus is to develop classifiers that perform well on both metrics.

The current solutions to address class imbalance modify the number of observations by generating new instances (over-sampling, SMOTE) or removing some (under-sampling). While techniques tailored to specific classifiers or problem domains have been proposed in the literature (Batuwita and Palade, 2010, Wang, Minku, and Yao, 2015), to our knowledge, no solution exists to identify features that highlight the minority class. Also, while n-gram frequency counts of textual data are often used as features, a targeted frequency count approach as we propose has not been attempted before.

The concept of class imbalance extends to multi-class problems where the target variable can assume more than two distinct values and the distribution of these values in the dataset need not be uniform. Here again the metrics for assessing model performance needs to be more nuanced and selecting the right features can go a long way in yielding more optimal models.

3. Related Work

Interest in class imbalance learning (CIL) has seen a sustained growth in the last 15 years due to its wide applications and the challenges inherent in such problems. A comprehensive survey of this field can be found in the work by He and Garcia (2009). In particular, the authors survey the current state-of-the-art methods to tackle the CIL problem including sampling methods, cost-sensitive learning, kernel based learning and active learning methods. The downsides to the sampling methods are explained with steps to overcoming them (such as the *EasyEnsemble* and *BalanceCascade* algorithms, cluster-based sampling methods etc.). The authors also provide an in-depth overview of using support vector machines for kernel-based learning and their integration with active learning methods.

A closely related area to CIL is *class overlapping*. This is the phenomenon of observations with

similar (almost identical) feature values belonging to different classes. For such problems, the difficulty arises from creating models with good performance metrics since class boundaries are not well demarcated. This problem is exacerbated for imbalanced classes if the few observations belonging to the minority class lie in the region occupied by the majority class instances. By simulating clusters of datasets at different distances, Prati, Batista and Monard (2004) show that class overlapping provides an equal if not greater challenge than class imbalance when applying machine learning algorithms. A unified view of class overlap for imbalanced data is presented in (Santos et al. 2023) where the authors propose a novel taxonomy of class overlap measures and characterize the relationship between them. In particular, the authors divide the complexity measures for class overlap into three categories: decomposition of the data space, identification of problematic regions, and quantification of the overlap problem. This extends the work of Ho and Basu (2002) who propose several data complexity measures to classify the hardness of supervised learning problems. These measures are divided into the categories: measures of overlap of individual feature values, measures of separability of classes and measures of geometry, topology and density of manifolds (Ho and Basu, 2002). As we elaborate further in Section 7, our feature selection method can be used as a potential aid in alleviating the class overlapping problem when there is an abundance of features that can be extracted from the associated metadata of the problem.

Several researchers address CIL for specific classifiers. For example, Fuzzy support vector machines (FSVMs) are adapted to handle class imbalance through the introduction of FSVM-CIL (Batuwita and Palade, 2010). The limitations of undersampling by ignoring many class instances are addressed through two techniques: EasyEnsemble and BalanceCascade (Liu, Wu and Zhou, 2009).

A survey of different machine learning algorithms for the CIL problem is conducted by Sarmanova and Albayrak (2013). Another approach to the CIL problems through the use of one-class classifiers and a survey of such algorithms is conducted by Sotiropoulos, Giannoulis and Tsihrintzis (2014). CIL problems are also encountered in online learning algorithms and two improved learning strategies OOB and UOB are proposed (Wang, Minku and Yao, 2015). The limitations of some algorithms in improving the accuracy of the minority class at the cost of impaired accuracy of the majority class is addressed through a hybrid classifier (Soda, 2011). The benefits of using class imbalance methods to predict defects in software modules is investigated in depth by Wang and Yao (2013).

When CIL problems have textual data associated with them (as ours does), NLP techniques come in handy and several researchers have addressed this area. Classification of software test cases into dependent and independent tests is an instance of CIL (Tahvili et al. 2020). The authors solve such problems by first converting the test cases to numeric vectors using NLP techniques and then applying supervised learning for imbalanced datasets on the resulting vectors. Rivera et al. (2020) classify news articles in a local Spanish newspaper as traffic-related or not. The authors first convert the article into a vectorized data using bag-of-words and TF-IDF (term frequency-inverse document frequency) techniques. They then experiment with five different classification algorithms on the resulting dataset. The class imbalance is dealt with via

different sampling methods. The final classifier achieves a sensitivity of 0.86 (Rivera et. al. 2020).

Class imbalance in online data streams often goes hand in hand with concept drift and a comprehensive overview of the current research in this field is conducted by Wang, Minku and Yao (2018).

For a review of the state-of-the-art classification algorithms for multi-class imbalanced learning, see (Bi and Zhang, 2018). The authors further propose a new multi-class imbalance classification algorithm that combines prior methods for tackling class imbalance and validates their results on 19 public datasets. Multi-class imbalanced data classification algorithms have also been implemented in open-source (Zhang et al. 2019).

For implementation aspects on CIL, including model evaluation, sampling techniques, cost sensitive training and ensemble algorithms in Python, see Brownlee (2020).

4. The Feature Extraction Step

In the context of a typical service-provider organization, a *customer support ticket* (CST) is created for a work order. As the work order item progresses towards completion, various aspects of the delivery are assessed for contract compliance. The service-provider organization wants to avoid any non-compliance with the service-level agreement, but, while such situations are rare, there may be discrepancies between what was promised to the customer and what was delivered. In the case where the service-provider organization delivers more than what the contract specified, the organization's resources are over-expended. They cost the service-provider not only for delivering beyond the contract, but also for following up with the customer post-audit. Our study addresses *over-delivery* in such situations. Our main question is: Can we identify the red flags in the early stages of execution and save on the cost of over-delivery with machine learning?

We worked with a dataset of CSTs where only a small fraction (less than 1%) were labelled as over-deliveries. Our objective was an automated solution that would be able to classify an existing CST as an over-delivery or contract-compliant. The biggest challenge to creating a useful classification model was the class imbalance.

We implemented a machine learning model that had been trained on prior data to predict whether an incoming CST was contract-compliant or constituted an over-delivery. The solution used textual data (see Figure 3 for a sample)[1] present in the CST to extract features that would highlight its tendency to be an over-delivery. Also included in the training data were categorical features associated with a CST. Using these features as input, we classified the CSTs with a random forest classifier.

[1] Some customer-specific data have been anonymized to ensure client confidentiality.

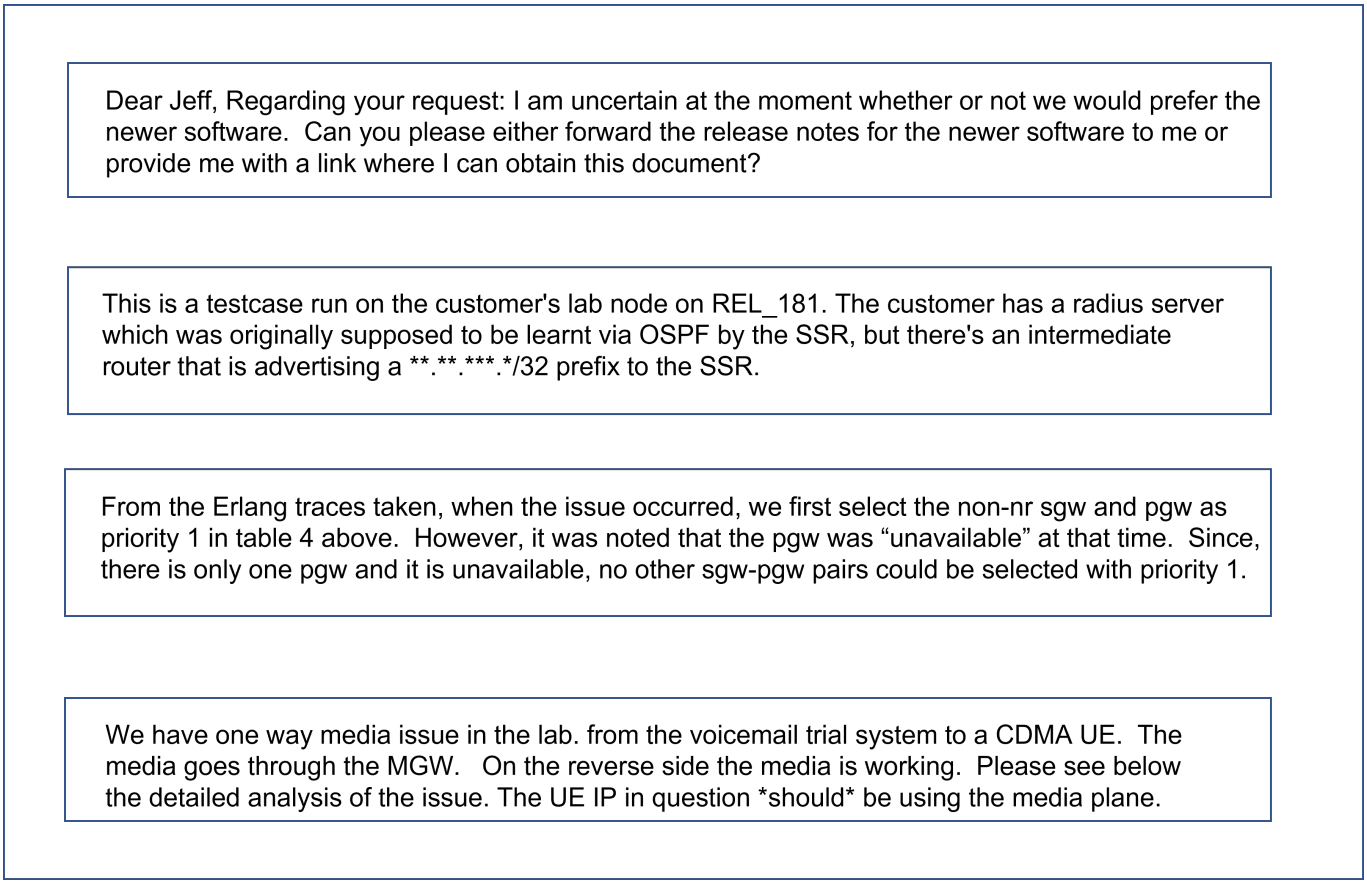


FIGURE 3: SAMPLE TEXT IN SOME CONTRACT-COMPLIANT (TOP 2) AND OVER-DELIVERED (BOTTOM 2) CSTS

Typically, the inputs for a machine learning modeling algorithm comprise of data fields already included in the dataset. In our case, we had a plethora of categorical features, some with many distinct values. We therefore performed a chi-square test of independence to select the subset of features that had the lowest p-values and therefore the highest correlation with the target variable. Some categorical features that were finally used in the model were business unit, competence domain, customer country id, customer type etc.

Also, while not implemented in our model, it is worth noting that the CSTs provide further opportunities for extracting out-of-band features by analyzing the metadata based on customer interactions vis-à-vis service level contracts.

To supplement the set of inputs for the classifier, we created additional features from the textual data inherent in the CSTs. These features are the frequency counts of specific n-grams in the text. The n-grams were identified using the concept of a *discrepancy score* which we define below.

Let N_{od} and N_{cc} be the number of over-deliveries and contract-compliant CSTs, respectively, in the training data.

Given an n-gram *w*, we define the *discrepancy score* disc(w) as the difference between the average frequency of occurrence of w in over-deliveries and contract-compliant CSTs. So, if w

occurs a total of x times across the N_{od} over-deliveries and a total of y times across the N_{cc} contract-compliant CSTs in the training data, then

(1) $$disc(w) = x/N_{od} - y/N_{cc}$$

Consequently, n-grams with high discrepancy scores will, on average, occur frequently in over-deliveries and rarely in contract-compliant CSTs. A high frequency count of an n-gram with a high discrepancy score is therefore indicative of an over-delivered CST.

Using this discrepancy score, we created new features from the textual data by performing the following steps:

1. Pre-process the textual data by removing all stop words, punctuation, and numerals
2. List out all the n-grams (in particular, unigrams, bigrams and trigrams) present in the CSTs
3. Calculate the discrepancy score of each n-gram using (eq. 1).
4. Identify the 20 unigrams, bigrams, and trigrams with highest discrepancy score (60 n-grams total)
5. Use the frequency counts of the n-grams identified in Step 4 as features.

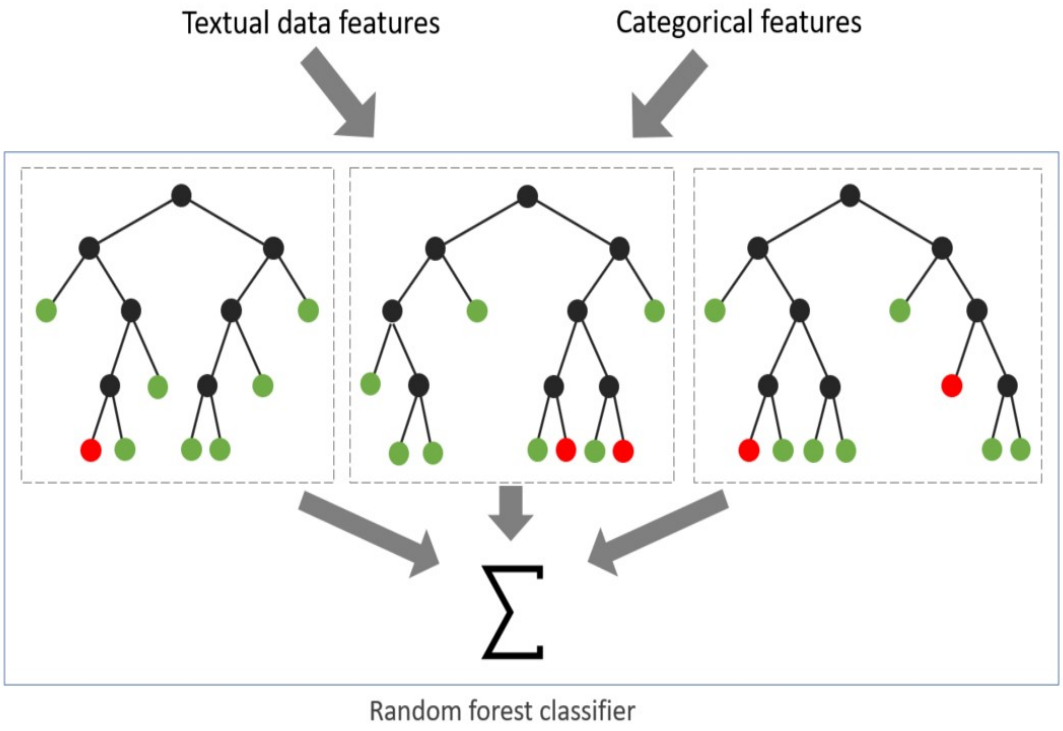


FIGURE 4. THE RANDOM FOREST CLASSIFIER MODEL BUILT WITH A COMBINATION OF CATEGORICAL AND TEXTUAL DATA FEATURES.

As illustrated in Figure 4, we inputted both the categorical feature values and the frequency counts of high-discrepancy n-grams into our model. The result was that the models that considered both categorial features and textual data features outperformed the models that only considered categorical features, as measured by precision, recall, and F_β-scores.

5. Implementation Results

To showcase the utility of the discrepancy score function, we implemented our algorithm on a collection of 31249 CSTs collected from different business units across Ericsson. Of this collection, 278 were labeled over-deliveries and the rest (30971) contract-compliant.[2]

We used a random forest classifier and performed a randomized grid search over the model hyperparameters. We preferred a randomized grid search over a complete grid search for the following reasons.

- As the hyperparameter space was large, a complete grid search to build the optimal model was impractical.
- Further, as the model had to be rebuilt for different scoring criteria (as we explain below), a repeated complete grid search would entail a prohibitive computational cost.

The tuned hyperparameters included the number of trees in the forest, the function to measure the quality of a split (gini or entropy) and the maximum depth of the tree. For the scoring function, we prefer models with high precision and recall values over accuracy so that naïve models that label all inputs as contract-compliant are discarded. In particular, we use the F_β scoring function which is a weighted harmonic mean of precision and recall (Brownlee 2020, 66-67).

Regarding model performance, the requirements from the stakeholder were:

- The precision of the resulting model should exceed a prescribed threshold so that its predictions of over-delivery are reliable.
- The fraction of over-deliveries detected by the model should be approximately equal to the actual fraction of over-deliveries in the dataset.

Taken together, what's desired is a model with good precision *and* recall. Keeping these two requirements in mind, we experimented with several models by choosing an F_β scoring function with $0.1 \leqslant \beta \leqslant 10$ (0.1 to 1 in steps of 0.1 and 1 to 10 in steps of 1). For $\beta < 1$, models

[2] For client confidentiality, our research was conducted on a sample of CSTs. These numbers do not reflect the true scale of contract-compliant or over-delivered CSTs handled by Ericsson.

with high precision (fewer false positives) are preferred while $\beta > 1$ favors models with high recall (fewer false negatives). Also in the randomized grid search, we use a different seed for each β so that the algorithm doesn't iterate over the same set of hyperparameters for different values of β. To avoid overfitting we use a cross-validation approach when building the model. However the cross-validation is applied directly on the dataset without introducing duplicate data instances. This will avoid the pitfalls of applying cross-validation as outlined in Section 1.

The final optimized model was found for $\beta = 5$ and satisfied the threshold for precision and declared over-deliveries.

To illustrate the value of using frequencies of n-grams with high discrepancy scores as features, we rebuilt the model in two alternate ways. In the first experiment, we again selected frequency counts of 60 n-grams as features (20 each of unigrams, bigrams and trigrams) but these n-grams were chosen randomly. The idea was to demonstrate the importance of discrepancy score by comparing two models of same feature complexity. In the second experiment, we built the model using only the categorical features.

Table 1 captures the computational results of our experiments. We observe that the model built using random n-gram frequency counts showed lower performance metrics reinforcing the benefit of using discrepancy scores for feature selection. Interestingly, the model built with only categorical features showed slightly higher precision but was much more conservative in calling out over-deliveries resulting in poor recall and F1-scores.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**% of declared over-deliveries**
Using both categorical features and frequency counts of n-grams with highest discrepancy scores	0.99	0.55	0.61	0.58	0.99
Using both categorical features and frequency counts of random n-grams	0.99	0.53	0.55	0.54	0.94
Using only categorical features	0.99	0.56	0.16	0.25	0.26

TABLE 1. COMPARISON OF RESULTS USING THE FEATURE SELECTION METHOD, RANDOM N-GRAMS AND ONLY CATEGORICAL FEATURES

Since the resulting metrics of the final model satisfy the stakeholder requirements, we decided to stop fine-tuning the model. But we note that it is possible to improve the model performance by performing one or more of the following steps:

- Expand the hyperparameter grid search space and increase the number of parameter settings that are sampled in the randomized grid search method.
- Select more n-grams with high discrepancy scores and use their frequency counts as additional features. For our experiment, we used 60 n-grams (20 each of unigrams, bigrams and trigrams with the highest discrepancy scores). One could choose more than 20 of these or choose higher order n-grams (4-grams, 5-grams etc.).
- Try other feature selection methods to identify n-grams whose frequency counts form well-separated distributions. We have outlined some of these methods in Section 7.
- Investigate other features that can be extracted from the textual data. The text accompanying each CST can also be used to mine further information such as
    - Number of engineers that worked on the CST.
    - The date of last modification of the CST.
    - The size of the textual data accompanying the CST.
    - Number of updates to the CST etc.

  Our feature selection methods can again be applied to select the best features resulting in improved models.

6. Extensions to Other Supervised Learning Problems

The underlying idea of extracting features from textual data can be extended to other classification problems (binary and multi-class). Consider the case where the target variable can take k possible values. We again assume that we have some textual data associated with each instance. The idea is to identify n-grams that highlight individual classes and use their frequencies as features. With each n-gram w, we associate the k-tuple $(x_1, \ldots, x_k)$ where x_i is the average frequency of occurrence of w among all observations belonging to the i^{th} class in the training data. We want to choose those w for which these x_i are most widely dispersed. For this, we define the discrepancy score for w as

(2) $$\text{disc}(w) = \min_{0 < i < j \leq k} |x_i - x_j|$$

Then we sort the n-grams by disc(w), choose the ones with the highest scores and use their frequencies as features. The n-grams selected by this method exhibit maximum separability in their frequency counts across the classes. Hence given an unlabeled observation, it is easier to deduce from the frequency values of these n-grams, the class it belongs to.

It is worth noting that the discrepancy score is one measure of dissimilarity between two

distributions. More generally, in a multi-class problem with k classes, we want to rank the (numeric) features by their ability to disambiguate the classes. The collection of all feature values belonging to a particular class in the training data forms a distribution and thus each feature results in a collection of k distributions. The feature importance is a measure of how well-separated these distributions are. If the distributions are clustered together, then given a new feature value from an unlabeled observation, it is hard to deduce which distribution generated this value. On the other hand, if the distributions are well separated, then given a new feature value, its easier to deduce the distribution it belongs to. Consequently the class, the observation belongs to, can be deduced.

For binary classification (k = 2), there are only 2 distributions and the discrepancy score we introduced in Section 4 is an estimate of the difference of the distribution means and thus a candidate for the dissimilarity measure (since the greater the difference, the more separated the distributions). This interpretation has the advantage of extending to all numeric features and not just frequency counts of n-grams. Other candidates for the dissimilarity measure are

- absolute value of the difference between the sample means (this would be an appropriate measure for non-imbalanced classes).
- difference between the sample medians.

A more general (and arguably more accurate!) measure of dissimilarity is the area of non-overlap between the two distributions (see Figure 5). Clearly, the more well-separated the distributions, the more the area of non-overlap.

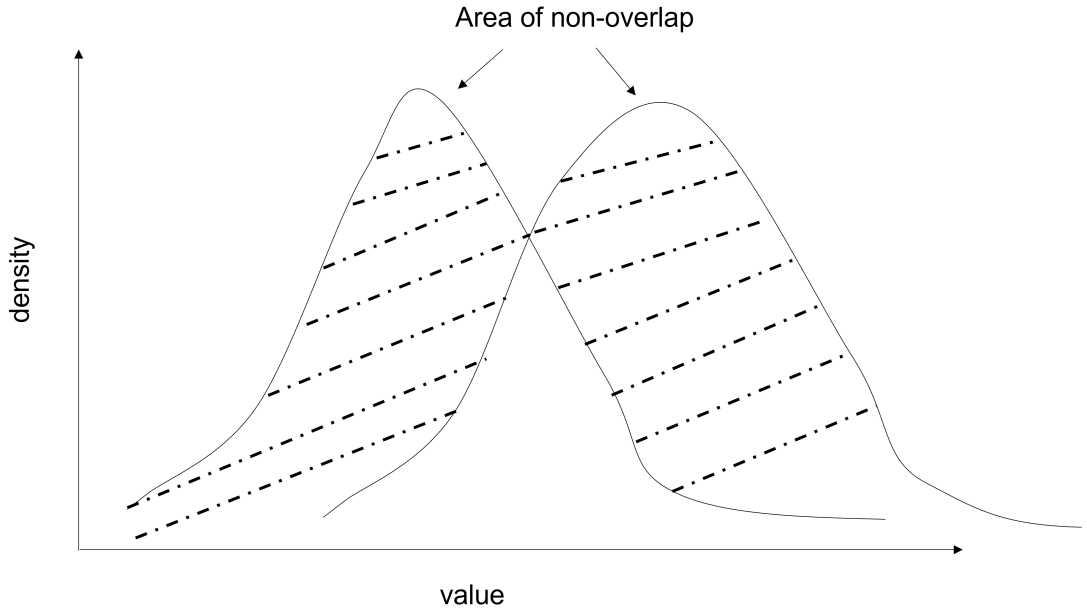


FIGURE 5. TWO DISTRIBUTIONS WITH THE AREA OF NON-OVERLAP SHADED[3]

[3] The area of overlap is a unitless quantity whose value lies between 0 and 2 since it aggregates two areas each of which is a portion of a probability density function (pdf) enclosing unit area.

With practical datasets, the feature values in the training data can be used to realize a normalized histogram which can be treated as an approximation to the underlying distribution. In our definition of discrepancy score, the sample mean was used as an estimate for the distribution mean.

For multi-class problems, the method we described above generalizes the solution in Section 4 for binary classification and identifies features whose distribution means are most separated. However a general measure of dissimilarity of k distributions would be desirable.

In a regression problem, the target variable is numeric. Since n-gram frequencies are numeric as well, a simple metric to gauge the strength of relationship is the correlation coefficient. In other words, we recommend choosing n-grams whose frequency of occurrence has the highest correlation with the target variable and use their frequencies as features.

7. Conclusions

In this article we addressed the class imbalance problem and showed how useful features can be extracted from the textual data inherent in many such problems. The utility of the features derived was measured in terms of the precision and recall values of the resulting classifier. Towards this end, we introduced the concept of discrepancy score that measures the effectiveness of a numeric feature in highlighting the minority class.

We demonstrated the efficacy of our method on a dataset of customer support tickets. We used both the categorical features inherent in a ticket as well as numeric features in the form of n-gram frequency counts that are inferred using the concept of discrepancy scores. In order to tune the model to meet the required precision and recall thresholds, we used the F_β scoring function for a range of values of β. To show the effectiveness of our method, we compared our model with one of the same feature complexity but where the n-grams selected for frequency counts were chosen randomly. We further constructed a model with only the categorical features. These had lower performance metrics, which strengthens our belief that the introduction of discrepancy scores leads to models with a useful balance of precision and recall.

We showed how our solution could be generalized and applied to other classification and regression problems. For a binary classification problem, the discrepancy score is an approximation to the difference of distribution means of the feature values belonging to the two classes and thus one representation of the dissimilarity of distributions. We argued that the area of non-overlap provides a more general measure of dissimilarity.

Frequency counts of n-grams in textual data is one illustration of how features can be derived from the metadata inherent in many datasets. The amount of metadata that can be derived is often quite voluminous and smart techniques are required to select the right features that can yield optimal models. Our solution and the subsequent generalization that we proposed provides some guidelines for selecting such features.

We also reviewed class overlapping and saw how this can be viewed as a potentially more serious problem than class imbalance. We believe that one way to overcome class overlapping is by considering more features with high discriminative powers. If there is sufficient metadata associated with the problem then the identification of additional features is feasible. The metadata can take the form of text or other tangential information related to the problem. Our feature extraction methods can then be used to detect discriminative features that can build more optimal models thereby reducing class overlapping.

Class imbalance learning finds applications in myriad domains and we believe that our feature selection approach to realize the optimal model can be extended to any problem with textual data or datasets containing a multitude of features.

Acknowledgment

We thank the referees for directing our attention to the class overlapping problem and other suggestions that improved the exposition of this paper. The first author dedicates his contribution to the memory of his late mother Smt. Vanaja Aravamuthan.

References

Batuwita, Rukshan, and Vasile Palade. 2010. "FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning." *IEEE Transactions on Fuzzy Systems* 18: 558—571. doi: 10.1109/TFUZZ.2010.2042721.

Bi, Jingjun, and Chongsheng Zhang. 2018. "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme." *Knowledge-Based Systems* 158: 81—93.

Brownlee, Jason. 2020. "*Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning*." Machine Learning Mastery. https://books.google.com/books/about/Imbalanced_Classification_with_Python.html?id=jaXJDwAAQBAJ

Chawla, Nitesh V. 2009. "Data Mining for Imbalanced Datasets: An Overview." In *Data Mining and Knowledge Discovery Handbook*, edited by O. Maimon and L. Rokach, Springer, Boston, MA. doi: 10.1007/978-0-387-09823-4_45.

He, Haibo, and Edwardo A. Garcia. 2009. "Learning from Imbalanced Data." *IEEE Transactions on Knowledge and Data Engineering* 21: 1263—1284.

Ho, Tin K., and M. Basu. 2002. "Complexity measures of supervised classification problems." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24: 289—300. doi: 10.1109/34.990132.

Liu, Xu-Ling, Jianxin Wu, and Zhi-Hua Zhou. 2009. “Exploratory Undersampling for Class-Imbalance Learning.” *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics* 39: 539—550. doi: 10.1109/TSMCB.2008.2007853.

Prati, Ronaldo C., Gustavo E.A.P.A. Batista and Maria C. Monard. 2004. “Class imbalances versus class overlapping: an analysis of a learning system behavior.” *4th Mexican International Conference on Artificial Intelligence*. LNCS, Mexico City, 2972: 312—321.

Rivera, Gilberto, Rogelio Florencia, Vicente García, Alejandro Ruiz, and J. Patricia Sánchez-Solís. 2020. "News Classification for Identifying Traffic Incident Points in a Spanish-Speaking Country: A Real-World Case Study of Class Imbalance Learning." *Applied Sciences* 10, 6253. doi: 10.3390/app10186253.

Santos, Miriam S, Jastin Pompeu Soares, Pedro Henriques Abreu, Hélder Araújo and João Santos. 2018. "Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]." *IEEE Computational Intelligence Magazine*, 13: 59—76. doi: 10.1109/MCI.2018.2866730.

Santos, Miriam S, Pedro Henriques Abreu, Nathalie Japkowicz, Alberto Fernández, and João Santos. 2023. "A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research." *Information Fusion* 89: 228—253. doi: 10.1016/j.inffus.2022.08.017.

Sarmanova, Akkenzhe, and Songül Albayrak. 2013. “Alleviating Class Imbalance Problem In Data Mining.” *21st Signal Processing and Communications Applications Conference (SIU)* 1—4. doi: 10.1109/SIU.2013.6531574.

Soda, Paolo. 2011. “A multi-objective optimisation approach for class imbalance learning.” *Pattern Recognition* 44: 1801—1810.

Sotiropoulos, Dionysios, Christos Giannoulis, and George A. Tsihrintzis. 2014 "A comparative study of one-class classifiers in machine learning problems with extreme class imbalance." *The 5th International Conference on Information, Intelligence, Systems and Applications* 362—364. doi: 10.1109/IISA.2014.6878723.

Tahvili, Sahar, Leo Hatvani, Enislay Ramentol, Rita Pimentel, Wasif Afzal, and Francisco Herrera. 2020. “A novel methodology to classify test cases using natural language processing and imbalanced learning.” *Engineering Applications of Artificial Intelligence*, 95, 103878, ISSN 0952-1976, doi: 10.1016/j.engappai.2020.103878.

Wang, Shuo, Leandro L. Minku, and Xin Yao. 2015. “Resampling-Based Ensemble Methods for Online Class Imbalance Learning.” *IEEE Transactions on Knowledge and Data Engineering* 27: 1356—1368. doi: 10.1109/TKDE.2014.2345380.

Wang, Shuo, Leandro L. Minku, and Xin Yao. 2018. "A Systematic Study of Online Class Imbalance Learning With Concept Drift." *IEEE Transactions on Neural Networks and Learning Systems* 29: 4802—4821. doi: 10.1109/TNNLS.2017.2771290.

Wang, Shuo, and Xin Yao. 2013. "Using Class Imbalance Learning for Software Defect Prediction." *IEEE Transactions on Reliability* 62: 434—443. doi: 10.1109/TR.2013.2259203.

Zhang, Chongsheng, Jingjun Bi, Shixin Xu, Enislay Ramentol, Gaojuan Fan, Baojun Qiao, and Hamido Fujita. 2019. "Multi-Imbalance: An open-source software for multi-class imbalance learning." *Knowledge-Based Systems* 174: 137—143. doi: 10.1016/j.knosys.2019.03.001.