# Classification of Life Events on Social Media

Paulo R. Cavalin
*IBM Research Brazil*
Rio de Janeiro, RJ, Brazil
e-mail: pcavalin@br.ibm.com

Fillipe Dornelas
*IBM Research Brazil*
Rio de Janeiro, RJ, Brazil
e-mail: fillipes@br.ibm.com

Sérgio M. S. da Cruz
*Universidade Federal Rural do Rio de Janeiro (UFRRJ)*
Rio de Janeiro, RJ, Brazil
e-mail:sergioserra@gmail.com

***Abstract*—In this paper we present an investigation of life event classification on social media networks. Detecting personal mentions about life events, such as travel, birthday, wedding, etc, presents an interesting opportunity to anticipate the offer of products or services, as well to enhance the demographics of a given target population. Nevertheless, life event classification can be seen as an unbalanced classification problem, where the set of posts that actually mention a life event is significantly smaller than those that do not. For this reason, the main goal of this paper is to investigate different types of classifiers, on a experimental protocol based on datasets containing various types of life events in both Portuguese and English languages, and the benefits of over-sampling techniques to improve the accuracy of these classifiers on these sets. The results demonstrate that a Logistic Regression may be a poor choice to deal with the original datasets, but after over-sampling the training set, such classifier is able to outperform by a significant margin other classifiers such as Naive Bayes and Nearest Neighbours, which do not benefit as well from the over-sampled training set in most cases.**

***Index Terms*—Life Events, Text Classification, Unbalanced Data**

## I. Introduction

Social Media Networks (SMN), such as Twitter, Facebook, Instagram and the like, engage thousands of people worldwide, which post a huge set of content on a daily basis [1], [2]. It is not uncommon that the content people post, such as a text, an image or a video, is intimately related to his/her personal life. In this category, we can cite the posting of content related to life events, which could not only be used to identify potential customers for a given product or service, but also to enhance their corresponding profiles in a given database.

In detail, a life event can be defined as something important that happened, is happening, or will be happening, in a particular individual's life. Some common life events are getting married, getting graduated, having a baby, buying a house, rellocating, getting a new job, and thus forth [3]. From a business perspective, the proper detection of life events could allow companies to anticipate the offer of products or services. For instance, if a person posts on the SMN that her wedding will be happening in a short time span (days or weeks, for example), a loan or an insurance for the couple's honeymoon trip could be offered in advance. And the chance to succeed tend to be higher in these cases because, as stated in [4], marketers know that people mostly shop based on habits, but that among the most likely times to break those habits, is when a major life event happens. In addition, detecting life events could also be a better way to estimate demographics and, consequently, to better understand a given population.

The literature demonstrates that applying machine learning methods is viable for detecting life events [4], [3]. Nevertheless, as demonstrated in [3], the recognition of a life event is categorized as an unbalanced classification problem, which means that the number of posts that does not contain life events is much higher than the number of posts that does. The main reason is that, besides the actual life events, a lot of non-personal content is generally posted on SMNs, such as advertisements, comments related to celebrities, jokes, and so forth. As a result, the training of a machine learning classifier to detect actual life events with high precision and recall rates, is challenging.

Given these standpoints, this work focuses in a broad analysis of classifiers for life event detection, with the following goals: 1) to define an experimental protocol, represented by datasets of distinct types of life events in different languages; 2) to better understanding the current state of this problem, by means of evaluating distinct machine learning classifiers on these datasets; 3) to evaluate the benefits of techniques to handle unbalanced datasets.

For achieving the aforementioned goals, the following methodology has been applied. We collected data from Twitter in two different languages, i.e. English and Portuguese, and selected and labeled binary datasets (positive samples are life events and negative ones are non life events) for six different types of life events, i.e. Travel, Weeding, Birthday, Birth, Graduation, and Death, with unbalance ratios, i.e. the proportion of positive versus negative samples, varying from 1.7% to 10.4%. Thus, by considering a standard text classification system based on bag-of-N-grams, we evaluated the performance of distinct types of classifiers on those sets, to get an overview of which category of machine learning classifier works best in this scenario: Naive Bayes (NB), a generative classifier; Logistic Regression (LR), which is a discriminative classifier; and Nearest-Neighbours (NN), which is instance-based. We then applied the same classification algorithms on datasets over-sampled by two different methods, i.e. random over-sampling and the SMOTE algorithm [5].

The results on the original sets, i.e. the not over-sampled ones, demonstrate that either NB and NN perform very close, while LR is generally a poor choice of classifiers. However, when the same machine learning classifiers are trained on over-sampled datasets, we observe a very significant increase of

performance for LR, making this type of base classifier achieve better results than the other two. Given that over-sampling generally results in higher increases in the Area Under the ROC Curve (AUC) score, the main metric for model selection considered in this work, we present a further investigation of the impact of ratio of increase of the set of samples of the minority class. These experiments demonstrate that increases of about 18% (on average, up to 26%) can be observed in AUC with the proper balance of the samples in the positive class, when training a LR classifier. Compared with the best results on the original sets, considering the three types of classifier, a LR classifier trained on an over-sampled training set can achieve 15% higher AUC scores, but it can be up to 25% depending on the set. In addition, it is worth mentioning that the techniques that have been used for this work, can be employed in other domains such as video and image analysis, with the appropriate adjustments.

## II. Related Work

As already mentioned, a life event can be defined as something important regarding the users' lives, which is posted on a SMN. It is important to differentiate it from some related work which uses the *event detection* expression to refer to the problem of detecting unexpected event exposed by several users in SMNs like a rumor, a trend, or emergent topic [6]. In our case, detection means to classify a short post, like Twitter's or Facebook's (two very popular SMNs in present day) status messages in one of the life event categories, which could be considered, for instance, similar to the topic classification problem, where each topic corresponds to a life event.

Research papers that focus on solving specifically the life event detection problem can be found in [7], [8], [9], [10], [11]. The type of data these approaches deal with can be divided into: 1) individual posts [7], [8], [9]; and 2) conversations or directly-linked sets of posts [10], [11]. While the latter can provide means for a more holistic understanding of the life events, i.e. understanding the *whats*, *whos*, *whens* and *wheres* of it, and possibly be more accurate in finding the true-positive cases, this type of data needs more complex approaches and datasets to model the relation between the messages in the conversation. For the former, on the other hand, one could apply more straight-forward natural language processing (NLP) approaches, making it simpler to develop a system for this case. In addition, individual-post life event detection allows the development of near-to-real-time reactive systems, since the text can be evaluated as soon as it is posted on the social media.

Regarding individual-post life event detection, both rule and machine-learning based systems have been problem. One example of the former is the system presented in [7], which has been employed for enhancing customers' profiles. Even though no measure of accuracy has been reported in that work, one clear disadvantage of the approach is the need to manually define rules, which can be costly to adapt the system to new life events and other languages. The work presented in [4], on the other hand, presents a machine-learning approach, which learns the life events from labeled sets of posts. They present the evaluation of different classifiers on two different life events, i.e. employment and marriage. Nevertheless, the datasets they consider are biased to positive examples, which drastically contrasts to what has been observed in real social media data, as reported in [3]. Furthermore, they do not take into account the problem of having to deal with multiple type of life events at the same time.

Given the issues mentioned in the previous paragraph, the system presented in [3] employs a hybrid method After the set of posts to be analysed is collected in the Ingestion phase, a rule-based method is applied in the Filter phase to find candidates for life events. The idea in this case, opposed to the method in [7], is to apply very simple rules, such as the main keyword that can point out a life event, just to reduce the search space for the subsequent phase. That is, by considering a set of life events, and their corresponding rules, the Filter phase associates to each post the list of possible life events, if there is any. Note that a lot of false positives may be present. Then, in the Detect phase, the binary classifiers, designed specifically for each life event, are applied on the posts to output a probability that the post actually contains the life event or not, only for those that appear in the list of life events. The method described herein, not only consists of a fast solution to be applied on large sets of posts, but also makes it easier to include new types of life events, as well as life events in other languages, compared with the systems in [7], [4].

## III. Datasets

Based on the system presented in [3], which proposes the creation of binary classifiers, one of each type of life event, in this section we describe the datasets that have been collected and labeled for life event classification. It is worth commenting that life event detection involves a multi-label classification problem, and a well-known solution for which is to the decompose the problem into a set of binary classifications [12].

The six types of live events considered in this work are:

- **Travel**: relates to people mentioning that they will travel somewhere in a short period of time;
- **Wedding**: when the person is going to get married soon;
- **Birthday**: when his/her birthday is happening or will be happening soon;
- **Birth**: when a son or daughter is going to be born soon;
- **Graduation**: when the person or a close relative is graduating at college;
- **Death**: when a close relative passes away.

We have collected eight different datasets from Twitter, by making use of the public Twitter Search API [13]. For two types of life events, i.e. Travel and Wedding, we have collected data in both English and Portuguese. For the remaining four types of life events, i.e. Birthday, Birth, Graduation and Death, only data in Portuguese has been collected, since we started this project first in Portuguese, but owing to time constraints we have not been able to carry out the labeling of the same

events in English yet. The list of datasets, with the respective keywords used to crawl the data, is presented in Table I.

Table I
THE LIST OF LIFE EVENTS AND THE CORRESPONDING KEYWORDS USED TO CRAWL THE DATA FROM TWITTER.

| Life event | Language | Search keywords |
|---|---|---|
| Travel-EN | EN | trip, travel, traveling, travelling |
| Wedding-EN | EN | marriage, married, marrying, wedding |
| Travel-PT | PT | viajar, viagem, viajando |
| Wedding-PT | PT | casamento, casório, casar |
| Birthday | PT | aniversário |
| Birth | PT | nascimento, nasceu, nascer |
| Graduation | PT | formatura, me formar, baile de formatura |
| Death | PT | faleceu, falecimento, morreu, morte |

The aforementioned datasets have been pre-processed in order to remove duplicates and retweets, and submitted to a labeling process, which engaged about five people. Since each life event has been labeled separately, the users just need to mark the posts that contained positive samples of the corresponding life events.

In Table II, we list the datasets, along with the number of samples of each, for both positive and negative cases. We can observe from the balance ratio (last column in the table), that these datasets are significantly unbalanced. At best, the total of positive samples corresponds to only 10.4% of the total of negative examples, as in the Birthday dataset. In the worst case, the positive examples can correspond to only 1.7% of the total of negative samples, such as in the Weedding-PT dataset. On average, that ratio is of about 5.0%. It is worth mentioning, also, that the size of the datasets are of about 1,825 samples, on average.

Table II
THE NUMBER OF POSITIVE SAMPLES (#P), NEGATIVE SAMPLES (#N), THE TOTAL OF SAMPLES (#TOTAL), THE POSITIVE/NEGATIVE BALANCE RATIO (%BALANCE), AND THE VOCABULARY SIZE $|\mathbb{V}|$, FOR EACH DATASET

| Dataset | #P | #N | #Total | %Balance | $\lvert\mathbb{V}\rvert$ |
|---|---|---|---|---|---|
| Travel-EN | 75 | 1925 | 2000 | 3.9 | 5,685 |
| Wedding-EN | 57 | 1943 | 2000 | 2.9 | 5,251 |
| Travel-PT | 109 | 1601 | 1710 | 6.8 | 4,154 |
| Wedding-PT | 44 | 2520 | 2564 | 1.7 | 5,320 |
| Birthday | 139 | 1339 | 1478 | 10.4 | 3,192 |
| Birth | 54 | 1561 | 1615 | 3.4 | 4,114 |
| Graduation | 95 | 1457 | 1552 | 6.5 | 2,096 |
| Death | 33 | 1646 | 1679 | 2.0 | 4,253 |
| Average | 76 | 1636 | 1825 | 5.0±2.9 | 4,258 |

## IV. METHODOLOGY

Given the life event datasets described in Section III, in this section we describe the methodology employed to evaluate the level of accuracy that can be achieved on these sets. For doing so, we first describe the baseline classification system, consisting of an standard approach to conduct text classification. Then, given that the datasets are very unbalanced, we describe the method we used to improve the performance of the system in unbalanced settings.

### A. Classification system

The baseline classification system consists of applying a standard approach for text classification, namely bag-of-words or bag-of-N-grams [14], after a few processing steps are carried out. In this work, the only two pre-processing steps applied are tokenization and stop-word removal.

In greater detail, consider the set of documents $\mathbb{D}$, where each document $d_i \in \mathbb{D}$ corresponds to a string of text, i.e. the original post. The tokenization phase consists of dividing each input text $d_i$ into the list of tokens denoted $\bar{t}_i$, where $\bar{t}_i \in \mathbb{T}$, containing the words or terms, punctuations, and other elements, such as URLs, hashtags, and emoticons (which commonly appear in social media posts), that appeared into $d_i$. Then, for each $t_i(j) \in \bar{t}_i$, if $t_i(j)$ also appears in the list of stop-word $\mathbb{SW}$, it is removed from $\bar{t}_i$, and we denote the new list as $\bar{t}'_i$, where $\bar{t}'_i \in \mathbb{T}'$ .

After obtaining the set $\mathbb{T}'$, the bag-of-N-grams features are extracted by computing the presence/absence of words and N-grams[1] in a previously-computed vocabulary $\mathbb{V}$. In other words, during this process, each list of tokens $\bar{t}'_i \in \mathbb{T}'$ is associated to a binary vector $x_i \in \mathbb{X}$ , where positions marked with 0 represent the absence of word $w_j \in \mathbb{V}$, while those marked with 1 represent its presence. Note that the dimensionality of $x_i$ is equal to the size of the vocabulary, i.e. $|x_i| = |\mathbb{V}|$. It is worth mentioning that $\mathbb{V}$ is generally computed from the set of tokens from the entire training set.

Then, the feature vectors $x_i \in \mathbb{X}$ can be used to both train and test a machine learning classifier. Given the high dimensionality of these vectors, as it can be observed from the vocabulary size list in Table II, the Naive Bayes classifier and linear classification methods, such as Logistic Regression, tend to perform well in text classification tasks. In addition, it is worth mentioning that, for each $x_i \in \mathbb{X}$, there is a corresponding class label $y_i \in \mathbb{Y}$, where $y_i$ is equal to one of the classes $\omega_k \in \Omega$.

### B. Methods to Deal with Unbalanced Datasets

It can be found in the literature different methods to deal with imbalanced datasets [15]. The approaches generally employ under-sampling, over-sampling, or a combination of both. While under-sampling consists of removing samples from the classes with larger number of samples, generally referred to as majority class, the over-sampling consists of increasing the

[1]An N-gram consists of a concatenation of $N$ consecutive tokens

number of samples of the minority class, i.e. the class with less samples.

Since under-sampling throws away information that can be useful for training, it can be outperformed by over-sampling, which keeps all the original knowledge and create new samplesto enhance the set. For this reason, we investigate two different methods for over-sampling, i.e. random over-sampling and the SMOTE algorithm.

Random over-sampling (ROS) consists of a simple way to conduct the over-sampling of a given dataset, by means of including repetitions of existing samples of the minority class, with the goal of given more weight to these samples and consequently improving the learning of a machine learning model. This approach, frequently referred to as over-sampling by replication, can be useful for some types of classifiers, in special generative models, for discriminative classifiers, such an approach fails to improve the computation of any classification decision region.

The Synthetic Minority over-sampling TEchnique (SMOTE), is an algorithm that has been proposed to deal with the issues of over-sampling by replication [5]. The main idea of SMOTE is to create new synthetic samples for the minority class (or classes, if it is a multi-class problem), based on the combination of existing samples and a random perturbation. That is, for each existing sample of the minority class, the idea is to find the nearest neighbours of the samples, and for each nearest neighbour, to create a new sample by adding the difference to nearest-neighbour sample multiplied by random numbers.

The SMOTE algorithm is presented in Algorithm 1, worth mentioning that we have slightly modified the notation presented in [5] for the sake of simplicity. The main inputs of the algorithm are: the training set, represent by $\mathbb{X}$ and $\mathbb{Y}$, containing respectively the data points and their respective class labels; the target class $y_{target}$, which is the minority class for which we would like to increase the number of samples; and the number of neighbours $K$ that will be used to increase the samples for class $y_{target}$. To make it clearer, suppose $|\mathbb{X}^{target}|$ is the total of samples for class $y_{target}$, in the end of the algorithm, $K \times |\mathbb{X}^{target}|$ samples will be included into $\mathbb{X}$. Given this input, the algorithm basically works as follows. In step 2, the new temporary set $\mathbb{X}'$ is created. Next, in steps 3 to 12, the creation of new synthetic samples is done by means of finding the $K$-nearest neighbours of $x_i$, for each $x_i$ where $y_i = y_{target}$ , and for each nearest neighbour of $x_i$, denoted $x_k$, compute the difference, denoted $dif$, between the dimension $l$ of $x_i$ and $x_j$ (step 6). Next, compute a random number between 0 and 1, save it into $gap$ (step 7), then compute the position $l$ of the new sample $x'$ by adding $gap \times dif$ to $x_i(l)$, as in step 8. The new samples are inserted into the temporary set $\mathbb{X}$. Afterwards, in steps 13 to 16, all the new samples are appended to the original set $\mathbb{X}$, and the associated class labels, which in this case are equal to $y_{target}$, are appended to $\mathbb{Y}$.

**Algorithm 1** The main steps of the SMOTE algorithm.

1: **Input:** $\mathbb{X}$ and $\mathbb{Y}$, the set of samples and the set with their corresponding class labels; $y_{target}$, the target class; and $K$, the number of neighbours to increase the number of samples is increased for $y_{target}$
2: $\mathbb{X}' = \emptyset$
3: **for** each $x_i \in \mathbb{X}$, where $y_i = y_{target}$ **do**
4: **for** $x_k \in \mathbb{X}$, where $y_k = y_{target}$ and $x_k$ is one of the $K$-nearest neighbours of $x_i$ **do**
5: **for** $l = 1$ until $|x_i|$ **do**
6: $dif = x_k(l) - x_i(l)$
7: $gap = random\ number\ between\ 0\ and\ 1$
8: $x'(l) = x_i(l) + gap \times dif$
9: **end for**
10: Insert $x'$ into $\mathbb{X}'$
11: **end for**
12: **end for**
13: **for** each $x'_i \in \mathbb{X}'$ **do**
14: Append $x'_i$ to $\mathbb{X}$
15: Append $y_{target}$ to $\mathbb{Y}$
16: **end for**

## V. Experimental Evaluation

In the present the experimental evaluation presented herein, the main objectives are twofold. The first goal is to analyze the baseline performance of the system described in Section IV-A, considering three distinct types of base classifier, i.e. Naive Bayes (NB), Logistic Regression (LR), and Nearest-Neighbour (NN) (which simply consists of a K-Nearest-Neighbour classifier with $K$ set to 1, for the sake of simplicity), on the life event classification datasets described in Section III. The second objetive is to investigate the use of the over-sampling techniques described in Section IV-B, to increase the samples in the minority class, i.e. the class containing positive samples of life events, and to observe the impact of this in the overall performance of the system, represented by the Area Under the ROC Curve (AUC) score, which is indicated as an appropriate metric to evaluate binary and imbalanced datasets [16].

For these experiments, we consider both 1-gram and 2-grams as features, generating a single feature vector with the combination of both. And the implementation of the base classifiers is based on the Scikit Learn library[2]. Regarding the NB classifier, a Gaussian model is used.

Even though AUC is the main metric considered for comparing the classifiers, we also present the results of the well-known Precision (Pr), Recall (Re), F1 Score (F1), and Accuracy (Ac) [17] metrics for the sake of completeness. For computing all metrics, Leave-One-Out evaluation is taken into account. That is, for each sample $x_i \in \mathbb{X}$, we train a classifier with all samples in $\mathbb{X}$ except $x_i$, and evaluate the classification of this sample. This process is repeated for all samples, and a confusion matrix is built from the results of each sample.

[2]http://scikit-learn.org/

In Table III we present the comparison of the three aforementioned base classifiers, on the eight datasets described in Section III. The results highlight that accuracy is rarely a good measure of performance for unbalanced data, as already stated in [5]. Just in twos cases, i.e. Graduation-PT and Death-PT datasets, the best accuracy is associated to the best AUC. In some cases, such as the Birth set, the best accuracy is related to a classifier that present precision, recall, and F1 equal to null, and the AUC is 0.5, owing to the very low balance ratio of this dataset, where high accuracy can be reached by predicting all samples as negative, i.e. non life event. Regarding the performance of the different classifiers, it is surprising that NN seems the best choice of classifier, since it is not a widely used type of classifier in text classification, presenting the best AUC scores in six datasets. The second best was NB, winning in the two remaining sets. And LR always seens to be a poor choice, tying in the first place with NN in a single set.

Table III
RESULTS OF THE BASELINE CLASSIFICATION METHODS ON ALL DATASET, FOR THE THREE BASE CLASSIFIERS: NAIVE BAYES (NB), LOGISTIC REGRESSION (LR), AND NEAREST-NEIGHBOUR (NN), WHERE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

| Dataset | Clas. | Pr | Re | F1 | AUC | Ac |
|---|---|---|---|---|---|---|
| Travel-EN | NB | 0.28 | 0.35 | **0.31** | **0.65** | 0.94 |
| | LR | **0.45** | 0.07 | 0.12 | 0.53 | **0.96** |
| | NN | 0.13 | **0.40** | 0.20 | 0.61 | 0.88 |
| Wedding-EN | NB | 0.26 | 0.47 | **0.34** | **0.71** | 0.95 |
| | LR | **1.00** | 0.04 | 0.07 | 0.52 | **0.97** |
| | NN | 0.06 | **0.60** | 0.11 | 0.66 | 0.74 |
| Travel-PT | NB | 0.14 | 0.24 | 0.17 | 0.56 | 0.86 |
| | LR | **0.54** | 0.17 | 0.26 | 0.58 | **0.94** |
| | NN | 0.32 | **0.48** | **0.38** | **0.69** | 0.90 |
| Wedding-PT | NB | 0.10 | 0.23 | 0.14 | 0.59 | 0.95 |
| | LR | 0.00 | 0.00 | 0.00 | 0.50 | **0.98** |
| | NN | **0.12** | **0.34** | **0.18** | **0.65** | 0.94 |
| Birthday-PT | NB | 0.31 | **0.45** | 0.36 | 0.67 | 0.85 |
| | LR | **0.62** | 0.28 | **0.39** | 0.63 | **0.92** |
| | NN | 0.33 | 0.40 | 0.36 | **0.66** | 0.87 |
| Birth-PT | NB | **0.19** | **0.28** | **0.22** | 0.62 | 0.93 |
| | LR | 0.00 | 0.00 | 0.00 | 0.50 | **0.97** |
| | NN | 0.09 | 0.22 | 0.13 | **0.66** | 0.90 |
| Graduation-PT | NB | 0.26 | **0.41** | 0.32 | 0.66 | 0.89 |
| | LR | **0.68** | 0.21 | 0.32 | **0.60** | **0.95** |
| | NN | 0.53 | 0.24 | **0.33** | **0.60** | 0.94 |
| Death-PT | NB | 0.02 | 0.03 | 0.02 | 0.50 | 0.95 |
| | LR | **0.50** | 0.06 | 0.11 | 0.53 | **0.98** |
| | NN | 0.23 | **0.09** | **0.13** | **0.56** | **0.98** |

The same evaluation of classifiers described in the previous paragraph has been conducted with the training sets over-sampled with both random over-sampling and SMOTE, with $K$ set to 5 to increase the samples in the positive class, i.e. with the minority class set enlarged in 500%. The results are listed in Table IV, where the best results are highlighted in bold. Except for the Weeding-PT and Birth-PT dataset, we can observe gains on all sets. In addition, we note that NB does not benefit from over-sampling in none of them, even though that is not degradation either. With the NN classifier, gains are observed in six dataset, and loss in two (Wedding-EN and Wedding-PT). With the LR classifier, on the other hand, we see increased values of AUC in all datasets, with and average increase of 18%, which is about 9 percentage points high than the average increase in 9% of NN. By comparing ROS with SMOTE, we observe just a slight advantage of the latter over the former with LR, with an increase of 17% of the AUC. But with NN, the use of the former results in a loss of 1.5%, showing that the over-sampling technique to be selected in dependent on the type of base classifier.

Although, after over-sampling the training set, the average AUC score of LR is still below that of NN, i.e. 0.64 against 0.69, LR is the type of classifier that demonstrated to benefit the most from these new training sets. For this reason, in Figure 1 we present the results of an evaluation of impact of increasing the value of $K$ from 0 (the original dataset) until 40, with increments of 5, to better observe the impact of the ratio of over-sampling on such classifier. In this case, we consider only SMOTE given its generally better results as described in the previous paragraph. It is interesting that, on all datasets improvements in AUC scores for $K \geq 10$ can be observed, i.e. by increasing the size of the minority class in at least ten times. The best value of $K$ is 25 for Travel-EN; 30 for Wedding-EN, 35 for Wedding-PT, Birthday-PT, and Death-PT Travel-PT; and 40 for Travel-PT, Birth-PT, and Graduation-PT. and Birthday. Compared with the LR classifier training on the original set, i.e. $K = 0$, over-sampling can produce an average increase of 37.6% in AUC, varying from 31.6% (Graduation-PT) to 46.1% (Wedding-EN).

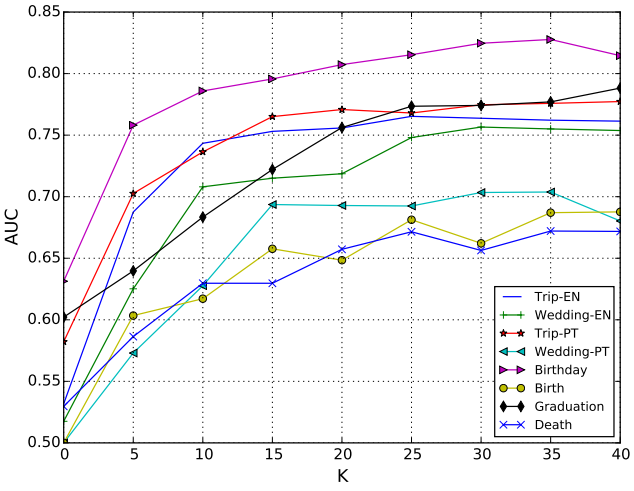


Figure 1. Evolution of the AUC score by varying SMOTE's $K$ parameter, from 0 to 40.

A summary of the best results obtained with the over-sampled training sets, along with the best results on the original training sets (where $K = 0$) is presented in Table V. The results show that, even though LR can be a very poor

Table IV
RESULTS OF THE BASELINE CLASSIFICATION METHODS ON ALL DATASET, FOR THE THREE BASE CLASSIFIERS: NAIVE BAYES (NB), LOGISTIC REGRESSION (LR), AND NEAREST-NEIGHBOUR (NN), TRAINED ON DATASETS OVER-SAMPLED WITH BOTH ROS AND SMOTE, WITH $K = 5$. IN BOLD ARE THE BEST RESULTS IN THIS TABLE AND THOSE FROM TABLE III, AND UNDERLINE ARE THE RESULTS WHERE IMPROVEMENTS COULD BE OBSERVED WITH OVERSAMPLING.

| **Dataset** | **Clas.** | ROS | | | | | SMOTE | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Pr** | **Re** | **F1** | **AUC** | **Ac** | **Pr** | **Re** | **F1** | **AUC** | **Ac** |
| Travel-EN | NB | 0.27 | 0.34 | 0.30 | 0.65 | 0.94 | 0.28 | 0.35 | 0.31 | 0.65 | 0.95 |
| | LR | **0.47** | 0.36 | 0.40 | 0.67 | 0.96 | 0.43 | **0.43** | **0.43** | **0.68** | **0.96** |
| | NN | 0.13 | 0.38 | 0.20 | 0.64 | 0.88 | 0.08 | 0.85 | 0.14 | 0.65 | 0.61 |
| Wedding-EN | NB | 0.26 | 0.47 | 0.33 | **0.71** | 0.94 | 0.26 | 0.47 | 0.34 | **0.71** | 0.95 |
| | LR | 0.26 | 0.29 | 0.27 | 0.63 | 0.95 | 0.23 | 0.32 | 0.27 | 0.62 | 0.95 |
| | NN | 0.06 | 0.59 | 0.10 | 0.66 | 0.72 | 0.05 | 0.75 | 0.10 | 0.65 | 0.58 |
| Travel-PT | NB | 0.13 | 0.23 | 0.17 | 0.56 | 0.85 | 0.14 | 0.24 | 0.17 | 0.56 | 0.86 |
| | LR | 0.40 | 0.41 | 0.40 | 0.68 | 0.92 | 0.41 | 0.44 | **0.43** | 0.70 | 0.92 |
| | NN | 0.28 | 0.44 | 0.34 | 0.68 | 0.89 | 0.14 | **0.76** | 0.24 | **0.74** | 0.68 |
| Wedding-PT | NB | 0.10 | 0.22 | 0.13 | 0.59 | 0.95 | 0.10 | 0.23 | 0.14 | 0.59 | 0.95 |
| | LR | **0.24** | 0.13 | 0.17 | 0.56 | 0.97 | 0.22 | 0.20 | **0.21** | 0.57 | 0.97 |
| | NN | 0.10 | 0.34 | 0.16 | 0.64 | 0.93 | 0.04 | **0.75** | 0.08 | 0.62 | 0.71 |
| Birthday-PT | NB | 0.30 | 0.45 | 0.36 | 0.67 | 0.85 | 0.31 | 0.45 | 0.36 | 0.67 | 0.85 |
| | LR | 0.48 | **0.58** | **0.53** | **0.75** | 0.90 | 0.48 | 0.57 | 0.52 | **0.75** | 0.90 |
| | NN | 0.35 | 0.38 | 0.36 | 0.65 | 0.87 | 0.04 | 0.52 | 0.08 | 0.71 | 0.76 |
| Birth-PT | NB | 0.18 | 0.27 | 0.22 | 0.62 | 0.93 | 0.19 | 0.28 | 0.22 | 0.62 | 0.93 |
| | LR | **0.34** | 0.18 | 0.24 | 0.59 | 0.96 | 0.33 | 0.22 | **0.27** | 0.60 | 0.96 |
| | NN | 0.10 | 0.27 | 0.15 | 0.59 | 0.89 | 0.09 | **0.78** | 0.15 | 0.70 | 0.71 |
| Graduation-PT | NB | 0.26 | 0.26 | 0.32 | 0.66 | 0.89 | 0.26 | 0.41 | 0.32 | 0.66 | 0.89 |
| | LR | 0.40 | 0.35 | **0.37** | **0.66** | 0.92 | 0.36 | 0.34 | 0.35 | 0.64 | 0.92 |
| | NN | 0.56 | 0.23 | 0.32 | 0.60 | 0.94 | 0.21 | **0.79** | 0.33 | **0.82** | 0.80 |
| Death-PT | NB | 0.02 | 0.03 | 0.02 | 0.50 | 0.95 | 0.02 | 0.03 | 0.02 | 0.50 | 0.95 |
| | LR | 0.29 | 0.15 | **0.20** | 0.57 | 0.97 | 0.25 | 0.15 | 0.19 | 0.59 | 0.97 |
| | NN | 0.25 | 0.12 | 0.16 | 0.55 | 0.97 | 0.10 | **0.48** | 0.16 | **0.66** | 0.90 |

choice of classifier in the original training sets, by adjusting the balance of the positive versus negative samples with over-sampling techniques, a considerable increase of performance can be achieved with such classifier. The average gain in AUC is of about 15%, but it can be as high as 25% as in the Birthday-PT dataset.

## VI. CONCLUSION AND FUTURE WORK

In this work we presented an evaluation of life event classification, focused on better understanding how different classifiers perform on such unbalanced data, and how the results can be improved with proper techniques to treat unbalanced data. By considering the original data, the analysis of three different classifiers demonstrated that both Naive Bayes and Nearest-Neighbours classifiers can be better choices than Logistic Regression at first. Nevertheless, with the use of an over-sampling technique to create less unbalanced datasets, the latter has shown to be able to achieve much better results. Thus, in order to achieve the best accuracy on such problem, a Logistic Regression classifier, trained on an over-sampled dataset, tends to be the best choice.

This work has inspired us to pursue different future research directions. The first is to better investigate techniques to deal with unbalanced data, for instance under-sampling and ensemble-of-classifiers based techniques. More important,

Table V
BEST RESULTS ON EACH DATASET, CONSIDERING THE LOGISTIC REGRESSION CLASSIFIER AND THE BEST VALUE FOR $K$, AND THE BEST RESULT FROM TABLE III, MARKED $K$ =NA.

| **Dataset** | $K$ | **%Bal.** | **Pr** | **Re** | **F1** | **AUC** |
|---|---|---|---|---|---|---|
| Travel-EN | 0 | 3.9 | 0.28 | 0.35 | 0.31 | 0.65 |
| | 25 | 97.5 | **0.29** | **0.59** | **0.39** | **0.77** |
| Wedding-EN | 0 | 2.9 | **0.26** | 0.47 | **0.34** | 0.71 |
| | 30 | 163.9 | 0.15 | 0.61 | 0.24 | **0.76** |
| Travel-PT | 0 | 6.8 | 0.32 | 0.48 | 0.38 | 0.69 |
| | 40 | 272.3 | 0.31 | **0.65** | **0.43** | **0.78** |
| Wedding-PT | 0 | 1.7 | 0.12 | 0.34 | 0.18 | 0.65 |
| | 35 | 61.1 | **0.24** | **0.43** | **0.31** | **0.70** |
| Birthday-PT | 0 | 10.4 | 0.33 | 0.40 | 0.36 | 0.66 |
| | 35 | 363.3 | **0.38** | **0.80** | **0.51** | **0.83** |
| Birth-PT | 0 | 3.4 | 0.09 | 0.22 | 0.13 | 0.66 |
| | 40 | 138.4 | **0.23** | **0.43** | **0.29** | **0.69** |
| Graduation-PT | 0 | 6.4 | **0.53** | 0.24 | 0.33 | 0.66 |
| | 40 | 260.8 | 0.36 | **0.65** | **0.46** | **0.79** |
| Death-PT | 0 | 2.0 | 0.23 | 0.09 | 0.13 | 0.56 |
| | 35 | 70.2 | **0.27** | **0.36** | **0.31** | **0.67** |

another direction is to investigate whether the same improvements can be observed with other types of feature sets, such as features based on word embeddings.

## REFERENCES


[1] K. Ehrlich and N. S. Shami, "Microblogging inside and outside the workplace." in *ICWSM*, 2010.

[2] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 591–600. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772751

[3] P. Cavalin, M. Gatti, and C. Pinhanez, "Towards personalized offers by means of live event detection on social media and entity matching," in *1st International Workshop on Social Personalisation (SP 2014)*, 2014.

[4] B. D. Eugenio, N. Green, and R. Subba, "Detecting life events in feeds from twitter," *2012 IEEE Sixth International Conference on Semantic Computing*, vol. 0, pp. 274–277, 2013.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622407.1622416

[6] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, 2013.

[7] M. Hernandez, K. Hildrum, P. Jain, R. Wagle, B. Alexe, R. Krishnamurthy, I. R. Stanoi, and C. Venkatramani, "Constructing consumer profiles from social media data," in *Big Data, 2013 IEEE International Conference on*. IEEE, 2013, pp. 710–716.

[8] J. Khobarekar, "Detecting life events using tweets," Ph.D. dissertation, University of Illinois at Chicago, 2013.

[9] S. Choudhury and H. Alani, "Personal life event detection from social media," in *25th ACM Hypertext and Social Media Conference*. CEUR, 2014.

[10] J. Li, A. Ritter, C. Cardie, and E. Hovy, "Major life event extraction from twitter based on congratulations/condolences speech acts," in *Proceedings of Empirical Methods in Natural Language Processing*, 2014.

[11] P. R. Cavalin, L. G. Moyano, and P. P. Miranda, "A multiple classifier system for classifying life events on social media," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, Nov 2015, pp. 1332–1335.

[12] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int J Data Warehousing and Mining*, vol. 2007, pp. 1–13, 2007.

[13] Twitter, "Using the Twitter Search API," 2016, [Online; accessed 25-May-2016]. [Online]. Available: https://dev.twitter.com/docs/using-search

[14] S. M. Weiss, N. Indurkhya, and T. Zhang, *Fundamentals of Predictive Text Mining*. Springer Publishing Company, Incorporated, 2012.

[15] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. Artif. Int. Res.*, vol. 19, no. 1, pp. 315–354, Oct. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622434.1622445

[16] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 233–240. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143874

[17] M. Sokolova, N. Japkowicz, and S. Szpakowicz, *AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation, pp. 1015–1021. [Online]. Available: http://dx.doi.org/10.1007/11941439_114