Sentiment Analysis and Deceptive opinion spam for Hotel reviews

Text Mining Project(732A92)

Andreas Stasinakis(andst745)

January 19, 2020

Abstract

Customer reviews are a very useful tool for the hotels in order to improve their facilities and increase their revenue. While Sentiment Analysis(Opinion mining) of these reviews is a common application of NLP, not many papers have studied the detection of another type of opinion spam known as "Deceptive opinion Spam". In this project, we first perform a sentiment analysis using an SVM model for positive and negative reviews. The evaluation of the model is done in a different dataset(reviews from a different source). We also perform a deceptive opinion spam(i.e classify the reviews as deceptive or truthful) starting with a model selection between Gradient Boosting, SVM and Logistic regression. We evaluate the optimal model(Logistic) and we compare its results with the paper "Negative Deceptive Opinion Spam" by Ott et al in order to comment whether or not we manage to increase the accuracy of the classifier. From the results presented, it can be said that Logistic regression surpass humans in terms of accuracy for deceptive opinion spam.

Keywords: Sentiment Analysis, Opinion mining, Deceptive spam, Classification, scikit-learn

Contents

1 Introduction

As a customer, choosing a hotel for vacation is always a hard task due to the thousands of different and interesting options. Most of the times though, the customer's final decision depends on the reviews of the hotels he or she is interested in(i.e the ones that satisfy specific requirements such as price, location etc.). Moreover, hotel reviews are also important for the hotels themselves. It is crucial for the hotel to get a feedback from the customer in order to fix problems which may occur(negative review) or keep the same strategy(positive reviews). In this way, the hotel could win the competition and increase its revenue. Due to the exponential growth of the data during the past decades, it is not easy to go through all these reviews and classify them as positive or negative manually. Therefore it could be useful for the hotels to automate the procedure of classifying positive or negative reviews.

But, how do we know if one review is not fake? In order for the hotels to increase their revenue, especially the "weak" hotels, it is a common scenario to hire employees to write fake positive(for their hotel) or negative(for competitors) reviews. That kind of problems are called Deceptive opinion spam and they described from Ott et al.(2011)[5] as

"fictitious opinions that have been deliberately written to sound authentic in order to deceive the reader"

We should mention here that those kind of reviews are different from the Disruptive opinion spam which are defined from Ott et al.(2011)[5] as

"uncontroversial instances of spam that are easily identified by a human reader"

For the two reasons explained above, we split this project into two parts. In the fist part we perform a sentiment analysis of positive and negative reviews. We evaluate the method using a test data from the original dataset and another dataset which comes from a different source. That will give us a better understanding of how our model generalizes.

The second part of this project is another classification problem. Using the same dataset as before, we try to classify the reviews as deceptive or truthful given their polarity(positive or negative). More specific, the model is trained both in positive and negative reviews but for the evaluation part two separate datasets are being used(one only positive and the other only negative). On top of that, a comparison with the "Negative Deceptive Opinion Spam"[4] paper is being done.

1.1 Related Work

For the deceptive opinion spam, we compare our results with the "Negative Deceptive Opinion Spam"[4] which is a sequel of "Finding Deceptive Opinion Spam by Any Stretch of the Imagination"[5]. In Ott et al.(2013)[4] a comparison between machine learning and human judges is done. The point of the paper, which is proved from the results, is that machines surpass humans in terms of deceptive opinion spam classification.

In order to achieve that, an SVM is used and it is compared with classification results from humans. Using cross validation, Ott et al.(2013)[4] used a lot of different combinations of the data. For example, an SVM only on the positive(or negative) reviews is trained and then tested on both polarities. As far as this project concerned, we are only interested on the results Ott et al.(2013)[4] obtained, when the SVM was trained in both polarities but tested it on each polarity separately.

2 Theory

During this project, two classification problems are discussed and 3 models are used in total. In this section we provide some background for each one of the models.

2.1 SVM

For the classification of the polarity, we used a powerful model called *Support Vector Machine(SVM)*. We have a binary classification problem of the polarity, which can only take 2 values(positive or negative). The target of the SVM is to classify a new n-dimensional point using an n-1 dimensional hyperplane. It is obvious that many hyperplanes may exist therefore the algorithm tries to maximize the separation(margin) between the two classes in order to minimize a loss function(i.e find the *maximum-margin hyperplane*). The power of SVM is that can also work for non linear problems using a trick known as *Kernel trick*. Therefore, the SVM maps the input in a high dimensional feature space where a hyperplance is used for the classification[6].

Given a set of M labeled training data $L = (x_i, y_i), 1 \leq i \leq M$, where $X_i \in R^n$ and $Y_i \in -1, 1$, SVM wants to estimate the weights W, for optimizing the function of the hyperlane which is $w^T X_i + b = 0$, where $Y_i = 0$ for all i and b is the bias. For that procedure, two new planes which depend on the support vectors are introduced. The positive plane($w^T X_i + b = 1$) and the negative one $w^T X_i + b = -1$. SVM tries to maximize the margin by minimizing the equation $\min \frac{1}{2}||w||^2$[6].

We present a graphical representation of the SVM as appeared in [6]. As we can see below, the SVM provides clear decision boundaries. Therefore, every new data point can be classified according to those boundaries.

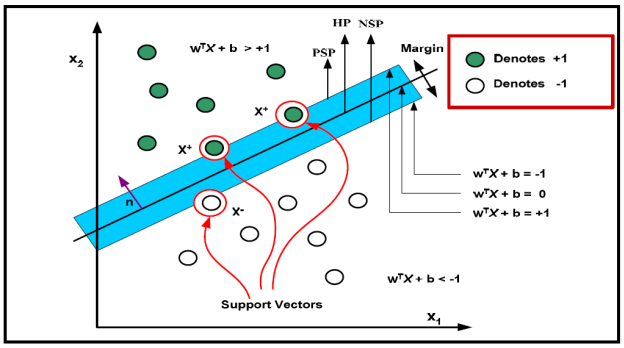

Figure 1: Graphical representation of SVM. X^+ and X^- are the positive and negative support vectors. PSP is the positive support plane, HP is the hyperplane and NSP is the negative support plane

Finally, SVM models tent to overfiting. For that reason a regularization parameter is added in the loss function in order to find a balanced trade-off between the maximum margins and the loss.

2.2 Logistic Regression

Despite the term Regression in the name, a *Logistic Regression* model is used for classification problems of several types such as binary or multinomial. Logistic regression models the probabilities of a class occur as a function of the predators. The final output of those probabilities is a linear combination of the predators x, which have been passed in a logit function before in order to ensure that the output is between 0 and 1 and its sum is 1[3, pp. 118-122].

To be more specific, assuming that we have a binary classification problem, where the response variable is $y = 0, 1$, and x is a matrix of the predators. Lets define as p the probability of class 1 given the data X(i.e $p = Prob(Y = 1|x)$).

The first step of the model is to model the logarithm of the odds. In the case of a binary problem $odds = \frac{p}{1-p}$. Therefore the

$$\begin{aligned} \ln \text{odds} = \text{logit(p)} = \ln \frac{p}{1-p} = b_0 + Bx \Rightarrow \\ \frac{p}{1-p} = e^{b_0+Bx} \Rightarrow \\ p = \frac{e^{b_0+Bx}}{1+e^{b_0+Bx}} \end{aligned} \tag{1}$$

where b_0, B are the coefficients we want to estimate

For fitting the model, the logistic regression uses the maximum likelihood estimation. The log-likelihood for the coefficients does not have a close form. For that reason, a gradient descent optimization is used[2].

2.3 Gradient Boosting

Gradient Boosting is a part of the Boosting algorithms. The idea behind Boosting is to fit weak classifiers and combine them in order to build a strong one and make more accurate predictions. A weak classifier can be defined as a classifier which performs slightly better than a random one[3, p.353-360].

The target of Gradient Boosting is to to minimize a pre-defined loss function using gradient decent. As a first step, a weak classifier(Tree) is trained. The residuals are calculated for the predictions of the weak classifier and a new weak model is fitted on those residuals. Numerical optimization(functional gradient decent) is used to minimize the loss function for the new model. Finally, the new model is added to the previous one. The procedure continues until a specific number of trees specified by the user is added.[1]

Therefore, the final "strong" classifier is an additive model which has been built sequentially through the above procedure[3, p.359-362].

3 Data

3.1 Deceptive Opinion Spam Corpus

Data used in this project has been downloaded from https://www.kaggle.com/rtatman/deceptive-opinion-spam-corpus. It is one of the very first datasets which includes *gold - standard* deceptive reviews. It was created for research purposes and more specifically for papers [5] and [4]. Ott et al. (2011) [5] created half of the dataset including *only* positive reviews and Ott et al. (2013) [4] the second half of *only* negative reviews. For this project, the combination of those two dataset is used.

The corpus contains data of 1600 rows and 5 columns for the 20 most popular hotels in Chicago. We are only interested in the three following columns: *reviews*, *polarity*(positive or negative) and *deceptive*(deceptive or truthful). The dataset is totally balanced and the labels of the polarity and deceptive columns are splitted as described below.

- 400 truthful positive reviews from TripAdvisor (created by Ott et al. (2011) [5])
- 400 deceptive positive reviews from Mechanical Turk (created by Ott et al. (2011) [5])
- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp (created by Ott et al. (2013) [4])
- 400 deceptive negative reviews from Mechanical Turk (created by Ott et al. (2013) [4])

We split the data into training(80%) and test(20%). We also used a 5-fold Cross validation on the training data for training the models.

In order to have a picture about the difficulty of the second classification problem(deceptive or truthful), we present two samples of the deceptive column, one for deceptive review and one for truthful, where both of them are positive reviews. It is clear that classifying those reviews seems tricky for a human being. Given that we do not present the gold class here, could you be sure which one is deceptive and which one truthful?

```
## FIRST REVIEW
##  My three night stay at Fairmont Chicago Millennium Park was the perfect way to end a weekend.
## The hotel has a classy but warm atmosphere with color tones that please the mind and soothe
##  the eyes.The staff was ready to help at any notice and always greeted you with
##  a cheerful smile. The mySpa was the getaway of a life time.
## My only complaint would be that the bed covers were a bit too stiff
##  as they were new and clean but the softness of the bed whisked me away to a deep slumber
##  in no time. Well worth the money and I know my next stay will be much longer.
```

```
## SECOND REVIEW
##  My stay at the The James was perfect. My room was thoughtfully designed:
## the lighting, the storage space, the bar area, the eating nook and everything else in the room
##  was purposeful and pretty to look at.
## The staff was friendly and and genuinely accomadating.
## The restaurant (Primehouse) echoed the above.
## Great service and one of the best steaks I've had in a very long.
```

3.2 Reviews data with ranking

The first problem which is addressed in this project is the classification of the polarity for each review. Due to the popularity of that task, we were able to find a different dataset in order to see how our model generalizes using a dataset which has been formulated from different sources. We name this dataset "external" and we are *only* going to use it in the first part of the project for *evaluation*.

The dataset is a sample of 1000 hotels, from December 2018 to May 2019, from a larger dataset provided by Datafiniti Business data. Among many variables, we are only interested in the "reviews" and the "ranking". The range of the ranking is between 0 and 5. In order to create the same labels as in the first dataset, we used the criteria below:

- *polarity* : " positive" if ranking ≥ 2.5

- *polarity* : " negative" if ranking < 2.5

The proportion of the negative reviews is barely close to 10% of the total reviews therefore the dataset is unbalanced. We could transform it to a balanced dataset using undersampling[8], but we decided to use it just as it is. Unbalanced data are very common in real-world applications, so it is interesting to see how our model performs in a unbalanced data from a different source.

3.3 Pre-processing

The dataset has a very good quality, without any missing values. Therefore a very simple preprocessing is done using SPACY package. During the preprocessing step, for each review, the lemma of each word in lower case *without* stop words, numbers and special characters is returned. Due to the small size of the data we did not use any threshold for the size of each review. As a result, all the reviews has been used despite their length.

4 Methods

4.1 Vectorizer

For both classification problems, we use the TF-IDF vectorizer from the package *sklearn*. Tf-idf computes numerically the importance of one word in a document. More specific, it is a combination of two terms[7]:

- TF(Term frequency) as a function of t(term) and d(document): This is simple the frequency of a term t in a document d. For example, if we have a document with 100 words and the word "madness" occurs 5 times, the $TF = \frac{5}{100} = 0.05$.
- IDF(Inverse Document Frequency) as a function of term t: This measures the frequency of a term in all the documents using the formula

$$IDF(t) = \log \frac{\text{N = Total number of documents}}{\text{Total number of documents that term t occurs}}$$

  Using the example above, assuming that $N = 10000$ and "madness" occurs in 100 documents then $idf(\text{"}madness\text{"}) = \log \frac{10000}{100} = 4.6$. In that way, we give more importance in words that do not occur that often. More specific, the TF for the term "a" would be probably be big. But the importance of "a" in a document is low. Therefore, IDF weights each term in order to give more importance in words that occur less than others.

Finally, the final vectorizer is $tf - idf(t, d) = tf(t, d) \cdot idf(t)$. The formula that TF-IDF from scikit-learn uses is slightly different but the original TF-IDF has the one presented above.

We should also mentioned that one parameter of the vectorizer is the *ngram_range*, which denotes the length of a sequence of words in a document. This sequence can be later seen as a feature of the data. In this project, we try Unigrams, Bigrams and Trigrams. Assuming that we have the document "Text Mining rocks", then for unigrams all three words("Text","Mining","rocks") are vectorized separately, while for bigrams and trigrams, pairs and triples are also modeled. More specific for Bigrams, also "Text Mining" and "Mining rocks" and for trigrams the entire "Text Mining rocks" are being used. This can be helpful in order to capture the meaning of a sentence, because the meaning of a sentence is lost when a bag of words approach is used. For large datasets the number of features is increasing and it may cause computational problems, but the dataset for this project is relatively small.

4.2 Binary Classification of the polarity using SVM

4.2.1 Uniform Baseline

The accuracy or any other numerical measurement is not always enough for the evaluation of a classifier. For that reason, in the first classification problem, we used a "dummy" model as a baseline. The baseline is a part of sklearn package and the function DummyClassifier was used. The model is trained on the training dataset, using the parameter "uniform". In this way, the baseline makes prediction uniformly at random.

4.2.2 SVM classifier with optimal parameters

For the polarity classification of the hotel reviews data we used an SVM model. Given the limited amount of data we have, a grid search of a 5-fold cross validation on the training data is used in order to train the model and tune the parameters.

A lot of combinations for different parameters have been evaluated on the training data, not only for the model but also for the vectorizer. The parameters, which gave the highest accuracy was the *linear SVM* for regularization parameter $C = 1$ and Bigrams as a parameter for the tf-idf vectorizer.

Finally, the generalization of that model was evaluated in two different ways. We first evaluated the model using the test data(20% unseen data from the original dataset). After that we used the second "external" dataset described in Data section in order to see how the model performs in a dataset from a different source.

4.3 Binary Classification of the deceptive data using Gradient Boosting, SVM and Logistic Regression.

The second part of the project is the classification of deceptive or truthful reviews. We train three models and after the model selection we tune the hyperparameters for the optimal model. Finally, a comparison of several evaluation criteria is done between our optimal model and the models from the paper "Negative Deceptive Opinion Spam"[4].

4.3.1 Human judges as a Baseline

In that task, we use human performance as a baseline. The results for that baseline were a part of the experiments in Ott et al. (2013)[4] for which three volunteer undergraduate university students were asked to classify the reviews as Deceptive or Truthful. Moreover, two meta-judges are also evaluated as classifiers. The *MAJORITY* meta-judge who classifies each review as deceptive when at least 2/3 human-judges classify it as deceptive as well and the *SKEPTIC* meta-judge who needs only one out of three to classify a review as deceptive. In this project, we use only the optimal result from Ott et al. (2013)[4] which is the majority meta-judge.

We should mention though that the data used for this "human baseline" are only 160 *negative* reviews. Therefore their answer can not directly be compared with the our classifier but they can be used as a baseline[4].

4.3.2 Model Selection

For training all three models, a 5-fold cross validation has been done on the training data using the default parameters. As we can see from the plot above, the logistic regression performs slightly better in terms of the average accuracy. Therefore, after the model selection, Logistic regression is the optimal model.

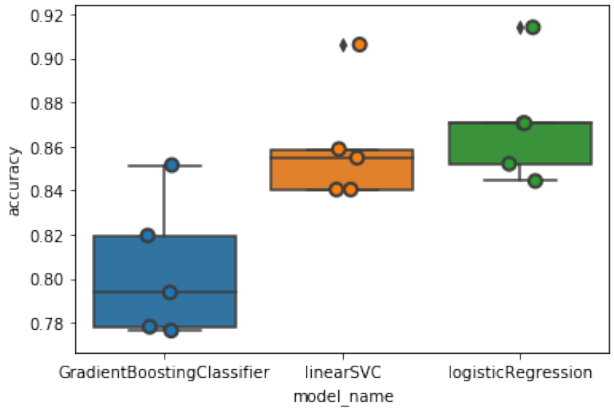

Figure 2: Boxplot of the accuracies for three models. A 5-fold cross validation has been done. Therefore, each bullet is the accuracy of the cross validation. The orizontal black line on each boxplot is the median of the accuracies for each model.

4.3.3 Logistic regression with optimal parameters

The logistic regression was only trained for the default parameters during the model selection procedure. In order to train the model and tune the parameters, a grid search of 5-fold cross validation on the training data is used again. The optimal parameters for the vectorizer is "ngram range" $= (1, 2)$(Bigram) and "binary" $= True$(set of words model). For the classifier, we checked different scales of the regularization parameter C and the optimal one was $C = 100$.

5 Results

5.1 Polarity classification

We first compare the accuracies of the baseline and the SVM classifiers on table 1. As mentioned before, for the SVM we have two different accuracies. As can be seen the accuracies of the two SVM models are higher compared to the baseline. Between the two SVM models, SVM_{test} performs better. We should mention again though, that the test dataset *is not* part of the dataset that the model has been trained.

Table 1: Accuracies of the two models. For the baseline and the SVM_{test}, we use the test data. For the $SVM_{external}$ we used the second dataset in order to compute the accuracy of the classifier

	Uniform Baseline	SVM_{test}	$SVM_{external}$
Accuracy	0.49	0.95	0.74

It is also interesting to compare the confusion matrices between the SVM models. On table 2, for the external dataset, the positive reviews were mostly misclassified(False negatives). On the other hand, on table 3, false positives are more.

Table 2: Confusion matrix for $SVM_{external}$.

	Predicted values	
Real values	negative	positive
negative	113	4
positive	257	626

Table 3: Confusion matrix for SVM_{test}.

	Predicted values	
Real values	negative	positive
negative	157	12
positive	5	146

5.2 Deceptive classification

The second part of this project is the deceptive or truthful classification using, after model selection, a Logistic Regression.

For a more objective interpretation we can compare the accuracies of the "metaJudge", "SVM_neg" and "LogR_neg". Therefore we compare the accuracies obtained from the test data with only negative reviews on table 4. It can be seen that both models have higher accuracies from the human judge. More specific, for the SVM the accuracy is 21% higher than the one of human judge(86 and 69.4 respectively), while for the Logistic regression that percentage is almost 24%.

Table 4: Evaluation criteria for meta-judge, SVM and Logistic Regression. More specific, P(precision), R(recall) and F(f1-measure). The training of the 2 models is done on the entire dataset. The polarity column defines the polarity of the test data. Therefore the evaluation is done in two different datasets, one positive reviews and one negative.

			Deceptive			**Truthful**		
	Polarity	Accuracy	P	R	F	P	R	F
metaJudge	negative	69.4%	68.7%	71.3%	69.9%	70.1%	67.5%	68.8%
SVM_pos	positive	88.4%	89.1%	87.5%	88.3%	87.7%	89.3%	88.5%
SVM_neg	negative	86.0%	86.7%	85.0%	85.9%	85.3%	87.0%	86.1%
LogR_pos	positive	88.0%	80.0%	97.0%	88.0%	97.0%	81.0%	89.0%
LogR_neg	negative	88.0%	83.0%	95.0%	89.0%	94.0%	79.0%	86.0%

6 Discussion

Looking at table 1 on section 5.1, despite the fact that for both datasets SVM surpass the "uniform" baseline (i.e therefore they generalize efficient), the SVM_{test} accuracy is almost 24% higher than the $SVM_{external}$. This can be easily explained though as we evaluate the model in a totally different dataset. First of all, we transform the problem from regression(rating column) to a binary classification problem. Moreover, the external dataset consists of reviews not only for hotels in Chicago but in US in general, which may lead to a greater variety of visitors. Therefore, many features of the model related to Chicago or the particular hotel "loses" its importance and leads to lower accuracy. Additionally, the external dataset is unbalanced. Even that though, the accuracy is still relatively higher(74%) than the baseline(49%).

For the confusion matrix of the external dataset, tables 2, it is obvious that the majority of the misclassified labels has a positive true label(False negative is 257 vs 4 as False positive). Again, we can say that this is happening because of the different datasets. In general for the FN and FP the data play a major role. Therefore, one explanation for the high number of FN is the fact that the external dataset is extremely *unbalanced*. Moreover, many reviews, for example positive, may have some negative aspects. But for the model we trained using the original dataset, these negative features are of high importance and lead to a misclassification for the external dataset.

For the deceptive opinion spam problem, a lot of conclusions can be made from the table 4 in section 5.2. The accuracies for the two models(SVM from Ott et al.(2014)[4] and logistic regression) are really close. For positive test data SVM is higher by almost 0.5%, while for negative reviews Logistic accuracy is greater than SVM by 2.3%. The most important conclusion though is, that the Logistic regression we trained performs much better than a human.

An interesting observation on table 4 is that for the SVM model in both test data the Precision, Recall and F1-measure are close to each other. The highest difference is less than 2% in terms of accuracy. On the other hand, for logistic regression, the range of the difference between the 3 metrics are high. The highest difference is 17% of accuracy and the lowest is 6%. Finally, if we assume that classifying a review as deceptive or truthful has the same importance(weight), we can use F1-measure for a direct comparison between the two models. For the positive test dataset, the difference between the two F1-measures are negligible. For negative reviews on the other hand, the F1-measure of Logistic regression for the deceptive class is almost 4% higher than the F1-measure from SVM.

7 Conclusion

In that project two different classification problems took place. The first task was simple enough as a result we were able to build a classifier with 95% test accuracy. On top of that we test that classifier in a dataset from a different source. Despite the fact that the accuracy was still much higher than the uniform baseline, a reduction of it has been occurred. That showed us that the way that a model generates depends a lot on the data. Small changes on a dataset may confuse the model, even if that model performs efficient enough in an unseen test data from the same source.

For the second part of the project, a more challenging classification problem was discussed. As we showed in the data section, it is quite hard for the two reviews to be classified correctly by human eyes. Given our results, we can comment that machines(in that case two different models) performs much more efficient in terms of accuracy from human beings. Between the two models, a general comment for which one is better cannot be done. In same cases, SVM performs better, while on some others the Logistic regression has higher metrics. Due to the small size and the good quality of the data, not many improvements directly to the model can be done. It would be interesting though, if a larger dataset exists in the future, to test some modern models, such as neural networks.

References

[1] Jason Brownlee. A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning Machine Learning Mastery. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[2] Scott A Czepiel. Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation. *Class Notes*, pages 1–23, 2012.

[3] Klaus Nordhausen. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition by Trevor Hastie, Robert Tibshirani, Jerome Friedman. *International Statistical Review*, 77(3):482–482, 2009.

[4] Myle Ott, Claire Cardie, and Jeffrey T. Hancock. Negative deceptive opinion spam. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference*, (June):497–501, 2013.

[5] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1:309–319, 2011.

[6] S V; Sathyanarayana, S; Amarappa. Data classification using Support vector Machine (SVM), a simplified approaCH. *International Journal of Electronics and Computer S cience Engineering, Volume 3, Number 4, ISSN- 2277-1956*, pages 435–445, 2014.

[7] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems*, 26(3):1–37, 2008.

[8] Show Jane Yen and Yue Shi Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. *Lecture Notes in Control and Information Sciences*, 344:731–740, 2006.