HARNESSING MACHINE LEARNING FOR INSIGHTFUL HOTEL REVIEW ANALYSIS

Utilizing Weka to predict ratings and reviewer influence from TripAdvisor data

Tripadvisor®

Table of Contents

Declaration of AI usage in this report

I hereby declare that all contents written and demonstrated below are completed by myself. I have used AI exactly as permitted in the assignment, that is for idea generation and proofreading.

1. Introduction

The prevalent online review has provided decision-makers for a long time how customers and travelers make decisions, particularly in the hospitality industry. Reviews on platforms like TripAdvisor or Airbnb offer real-time feedback into customer experiences. However, this presents a challenge in interpretation due to the unstructured format of the review. The problem at hand is twofold: first, to analyze the sentiment expressed in text reviews into meaningful overall ratings; and second, to tell apart the potential influence of a review based on its perceived helpfulness by others. Addressing these two problems is vital as they directly affect business reputation and service improvement strategies.

In this report, we are provided a dataset scrapped from online reviews on the TripAdvisor website regarding the hotel reviews. This dataset consists of 3,118 instances and 22 attributes, ranging from ratings, author information to hotel information. However, we will focus exclusively on textual information, so only these attributes below will be explored and analyzed by the Weka software.

Hotel_id	The id of the hotel being reviewed
id	The id of review
num_helpful_votes	The number of helpfulness votes that a review has received. This measures how influential a review is (numeric)
rating_overall	Customer's overall rating on this hotel in the review (nominal)
title	The title of a review (string)
text	The review text (string)

As mentioned above, the first task addressed in this report is to develop a predictive model for hotel ratings based on the content of review titles and text, while the second task involves creating a model to forecast the likelihood of a review being considered helpful by other travelers, as measured by receiving more than 15 helpful votes. Preliminary results from this report suggests a strong capability for the machine learning models to accurately reflect hotel ratings and to identify influential reviews, although careful model validation is necessary to ensure these models perform well on external data.

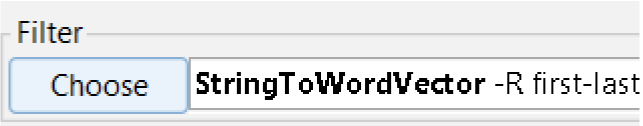
2. Problem definition and algorithm

2.1 Task definition

The problems we are addressing includes two modelling tasks:

Task 1: Modeling hotel ratings

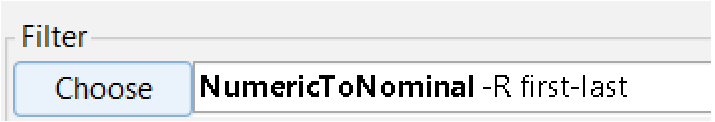
- Inputs: The features variables are the title and body text. They are transformed into word vectors using the unsupervised method, string to word vector in Weka. The word vectors are used directly for classification



Filter

StringToWordVector -R first-last

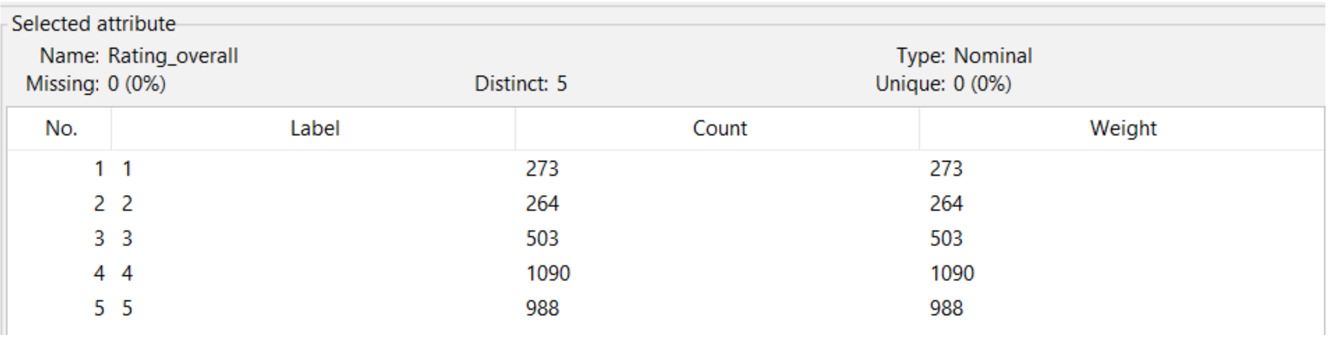
- Outputs: The output is the predicted overall hotel rating, which are nominal categorical variables ranging from 1 to 5 stars. Originally, the rating_overall attribute is in numeric data. However, for classification task, it is changed to a nominal attribute, using Weka's unsupervised filter



Filter

NumericToNominal -R first-last

As a result, this is a classification task which has 5 classes. The classes are quite balanced, so there is no need for oversampling in later stages.



Selected attribute

Name: Rating_overall
Missing: 0 (0%)
Distinct: 5
Type: Nominal
Unique: 0 (0%)

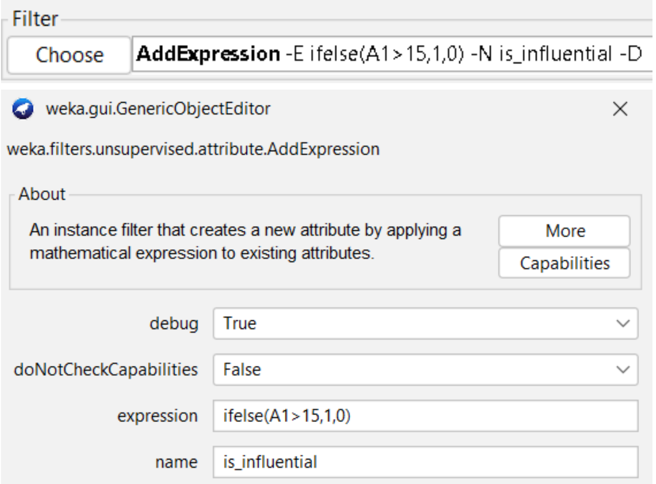
No.	Label	Count	Weight
1	1	273	273
2	2	264	264
3	3	503	503
4	4	1090	1090
5	5	988	988

- Importance of this task: This is a critical problem because accurate prediction of these ratings from text can help tell hotels their general performance. However, since this is just a predictive model on the overall rating, a more specific model trained on the service or cleanliness rating would be more actionable than this overall rating, which provides litle information on how the hotel could improve their ratings.

Task 2: Predicting review influence

- Inputs: Similar to the first task, inputs are the title and text of the reviews, which are then transformed into word vectors. However, it does not necessarily mean that the word vector transform configuration is the same as the first task. Specific settings of the word vectorizer would be provided later in the report.

- Outputs: The output is a binary attribute telling whether a review is influential or not. In other words, a new attribute is_influential is created based on the column num_helpful_vote, where is_influential is 1 if num_helpful_vote has at least 15 upvotes, and otherwise 0. Again, this is also a classification task like the first one.



AddExpression -E ifelse(A1>15,1,0) -N is_influential -D

weka.gui.GenericObjectEditor

weka.filters.unsupervised.attribute.AddExpression

About

An instance filter that creates a new attribute by applying a mathematical expression to existing attributes.

debug: True
doNotCheckCapabilities: False
expression: ifelse(A1>15,1,0)
name: is_influential

- Importance of this task: this problem is important as influential reviews can help businesses focus on feedback that directs customer decision making process the most. Influential reviews are interesting because they help explain what makes review content compelling and worthy of attention.

2.2 Algorithm definition

Below are the candidate algorithms and techniques that I would use in this report. For classification tasks, I would propose to use only 2 competing algorithms: the J48 decision tree and the random forest. A decision tree is a flowchart-like tree structure, where each internal node represents a test happening an attribute, each branch represents an ending of the test, class label is represented by each leaf node or terminal node. [1]

The stages of J48 decision tree algorithm [1]

1. The leaf is labeled with a similar class if the instances belong to the similar class.
2. For each attribute, the potential data will be figured out and the gain in the data will be taken from the test on the attribute.
3. Finally, the best attribute will be chosen depending upon the current selection parameter.

J48 algorithm is one of the best machine learning algorithms to examine the data categorically and continuously.

The Random Forest algorithm [2]

Random forest is one of the most popular classification algorithms due to its good performance on a small dataset and it is very resistant against overfitting. Generally, random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large.

The cost sensitive matrix [3]

Many classifiers are studied under the error-based framework, which concentrates on improving the accuracy of the classifier. On the other hand, the cost of misclassification is also an important parameter to consider in many applications of classification

In this report, I believe that the cost of misclassification is justifiable, because classifying 1 star review as having 2 stars would not be as serious as classifying 1 star review as having 5 stars for the first task. The same argument could be proposed for the second task, where missing influential reviews is often much more costly than mistaking non-influential reviews as influential.

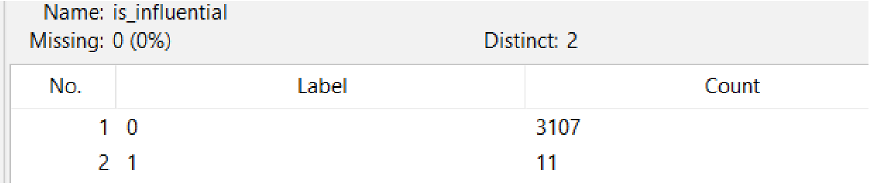
The information gain attribute evaluation [4] for feature selection

After conducting word vectors transformation on the title and text, we would easily have over a thousand attributes, and not all of them would be relevant for classification. As a result, we should also use some feature engineering techniques. The one that I used is **InfoGainAttributeEval** in the **SelectAttributes** tab in Weka, which evaluates the worth of an attribute by measuring the information gained with respect to the class.

InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute).

Synthetic Minority Over-sampling TEchnique (SMOTE) [5]

A dataset is imbalanced if the classes are not approximately equally represented. In Task 2, the influential and uninfluential reviews are extremely unbalanced, with uninfluential reviews 282 times greater than influential ones.



Name: is_influential
Missing: 0 (0%) Distinct: 2

No.	Label	Count
1	0	3107
2	1	11

SMOTE is an over-sampling approach in which the minority class is over-sampled by creating synthetic examples rather than by over-sampling with replacement. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. SMOTE should be applied after feature selection to reduce noise and computing power for new samples.

All algorithms and techniques above are available in Weka software.

3. Experimental evaluation

1. Criteria for evaluation

The classification performance is measured using accuracy, precision, recall, and F1-score across many different settings as follows

- Task 1 settings for both J48 and Random Forest:

Case 1.0: Baseline model with only word vectors
Case 1.1: word vectors + feature selection
Case 1.2: word vectors + feature selection + linear cost sensitive matrix
Case 1.3: word vectors + feature selection + nonlinear cost sensitive matrix

The cost sensitive matrix for task 1 is reported as follows

Linear cost matrix case

Actual ↓ /Predicted →	**1 star**	**2 star**	**3 star**	**4 star**	**5 star**
1 star	0	1	2	3	4
2 star	1	0	1	2	3
3 star	2	1	0	1	2
4 star	3	2	1	0	1
5 star	4	3	2	1	0

Nonlinear cost matrix case

Actual ↓ /Predicted →	**1 star**	**2 star**	**3 star**	**4 star**	**5 star**
1 star	0	1	2	4	8
2 star	1	0	1	2	4
3 star	2	1	0	1	2
4 star	4	2	1	0	1
5 star	8	4	2	1	0

- Task 2 settings for both J48 and Random Forest

Case 2.0: Baseline model with only word vectors
Case 2.1: word vectors + feature selection
Case 2.2: word vectors + feature selection + SMOTE for influential reviews
Case 2.3: word vectors + feature selection + SMOTE + cost sensitive matrix

The cost sensitive matrix for task 2 is reported as follows

Actual ↓ /Predicted →	**Uninfluential review (0)**	**Influential review (1)**
Uninfluential review (0)	0	1
Influential review (1)	100	0

2. Hypotheses tested

It is evident that the hotel ratings and the helpfulness of each review are highly correlated with the review text and review title. Additionally, I also hypothesize that using a cost sensitive matrix, conducting feature selection based on information gain and minority sampling for influential reviews will help improve the classification performance compared to baseline model.

3. Experimental methodology

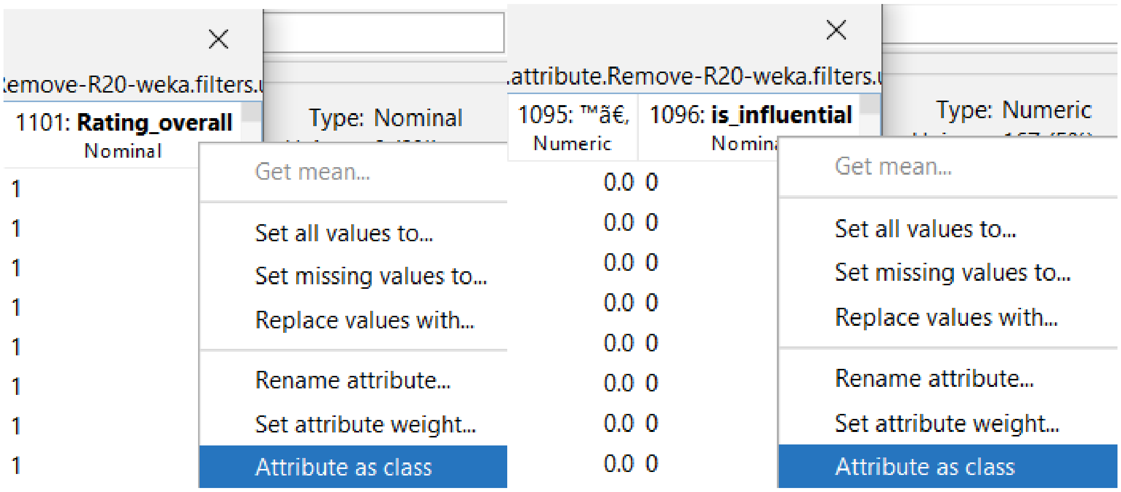
The analysis compares the performance of Random Forest and J48 classifiers. This comparison reveals how each modification influences the model's ability to classify reviews accurately. For performance evaluation, I used a standard train-test split to evaluate the model's performance, ensuring the model can perform well on unseen data. The training-testing ratio is 70/30

4. Dependent and independent variables

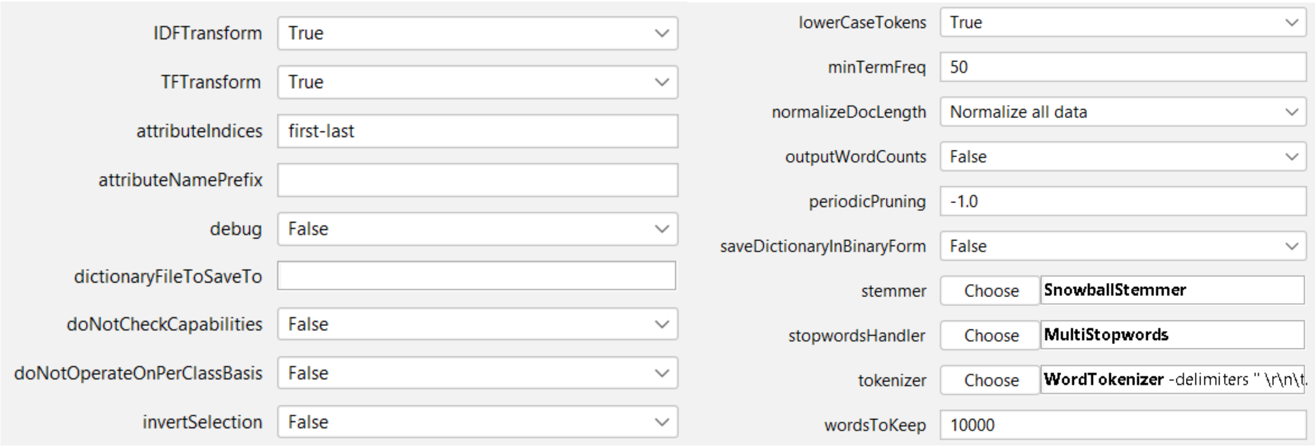
The dependent variable, again, is **Rating_overall** in nominal format for Task 1 and **is_influential** in nominal format, which is derived from num_helpful_votes attribute in nominal format in Task 2. Independent variables are derived text features from text and title of reviews, such as tokens and TF-IDF scores.

4. Results

First, the Rating_overall and is_influential are still attributes, so they should be converted to the class in the Editor for both tasks

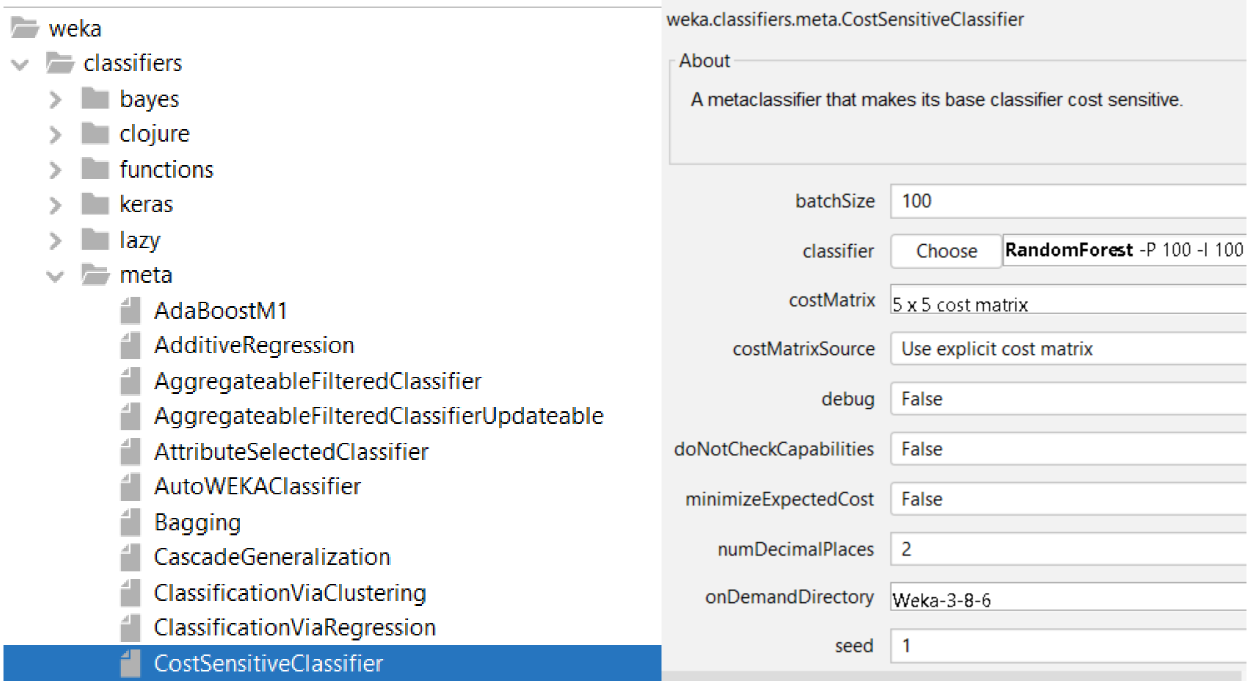


After that, the word vectors are created using this option, which is common for both Task 1 and Task 2. These options deliver the best classifications result for the baseline case using only word vectors.



IDFTransform	True	lowerCaseTokens	True
TFTransform	True	minTermFreq	50
attributeIndices	first-last	normalizeDocLength	Normalize all data
attributeNamePrefix		outputWordCounts	False
debug	False	periodicPruning	-1.0
dictionaryFileToSaveTo		saveDictionaryInBinaryForm	False
doNotCheckCapabilities	False	stemmer	Choose SnowballStemmer
doNotOperateOnPerClassBasis	False	stopwordsHandler	Choose MultiStopwords
invertSelection	False	tokenizer	Choose WordTokenizer -delimiters " \r\n\t.
		wordsToKeep	10000

Finally, for cost sensitive classification, we need to use the Cost Sensitive Classier as the classifying base in meta folder and choose either J48 or Random Forest as the classifier in the properties option. The cost matrix inside CostSensitiveClassier properties is for training process, while cost matrix inside Classifier evaluation options is to calculate the final cost on the testing dataset



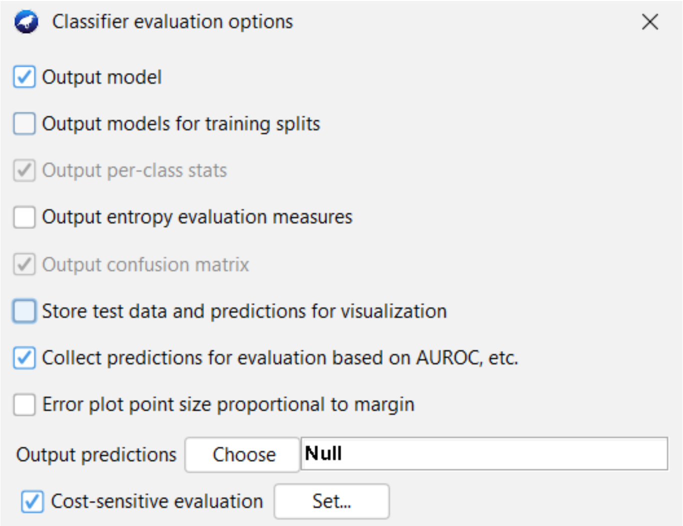
- weka
  - classifiers
    - bayes
    - clojure
    - functions
    - keras
    - lazy
    - meta
      - AdaBoostM1
      - AdditiveRegression
      - AggregateableFilteredClassifier
      - AggregateableFilteredClassifierUpdateable
      - AttributeSelectedClassifier
      - AutoWEKAClassifier
      - Bagging
      - CascadeGeneralization
      - ClassificationViaClustering
      - ClassificationViaRegression
      - CostSensitiveClassifier

weka.classifiers.meta.CostSensitiveClassifier

About

A metaclassifier that makes its base classifier cost sensitive.

batchSize	100
classifier	Choose **RandomForest** -P 100 -I 100
costMatrix	5 x 5 cost matrix
costMatrixSource	Use explicit cost matrix
debug	False
doNotCheckCapabilities	False
minimizeExpectedCost	False
numDecimalPlaces	2
onDemandDirectory	Weka-3-8-6
seed	1



Classifier evaluation options

☑ Output model

☐ Output models for training splits

☑ Output per-class stats

☐ Output entropy evaluation measures

☑ Output confusion matrix

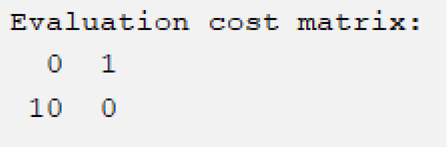
☐ Store test data and predictions for visualization

☑ Collect predictions for evaluation based on AUROC, etc.

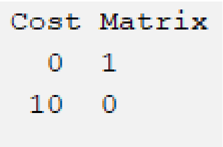
☐ Error plot point size proportional to margin

Output predictions Choose **Null**

☑ Cost-sensitive evaluation Set...



```
Evaluation cost matrix:
  0   1
 10   0
```



```
Cost Matrix
  0   1
 10   0
```

These two matrices can be different as well, but for consistency, they should be similar to measure the costs incurred by wrong classifications

4.1 Task 1 results

Baseline expectation: In a five-class classification task, random guessing would theoretically result in an accuracy of 20% (1 out of 5). Correctly predicting the exact star rating among five classes is much harder than binary classification.

In service industries like hospitality, a prediction that is off by one star might still be useful. For example, mistaking a 4-star for a 5-star review may not be as severe as mistaking a 1-star for a 5-star. Therefore, if most predictions are within one star difference from prediction, the model can be considered to perform decently well on unseen data.

After testing the four cases, the classification can be reported for J48 and Random Forest across accuracy, precision, recall and F1 score. For each score, we could compare side by side the four study cases

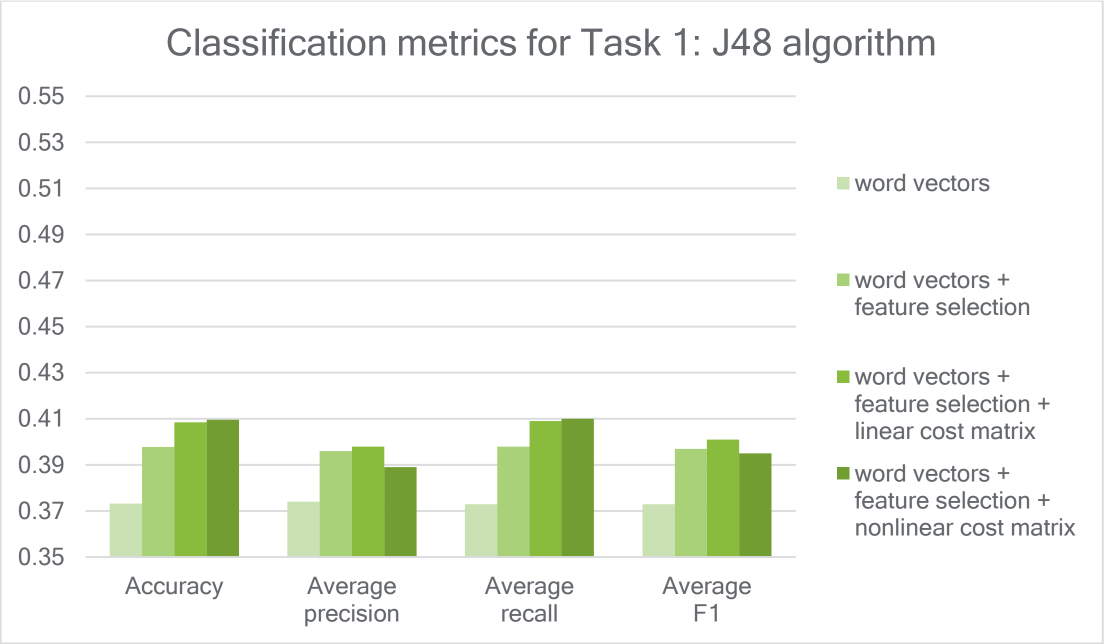


Figure 1: Classification results of Task 1 by J48 algorithm

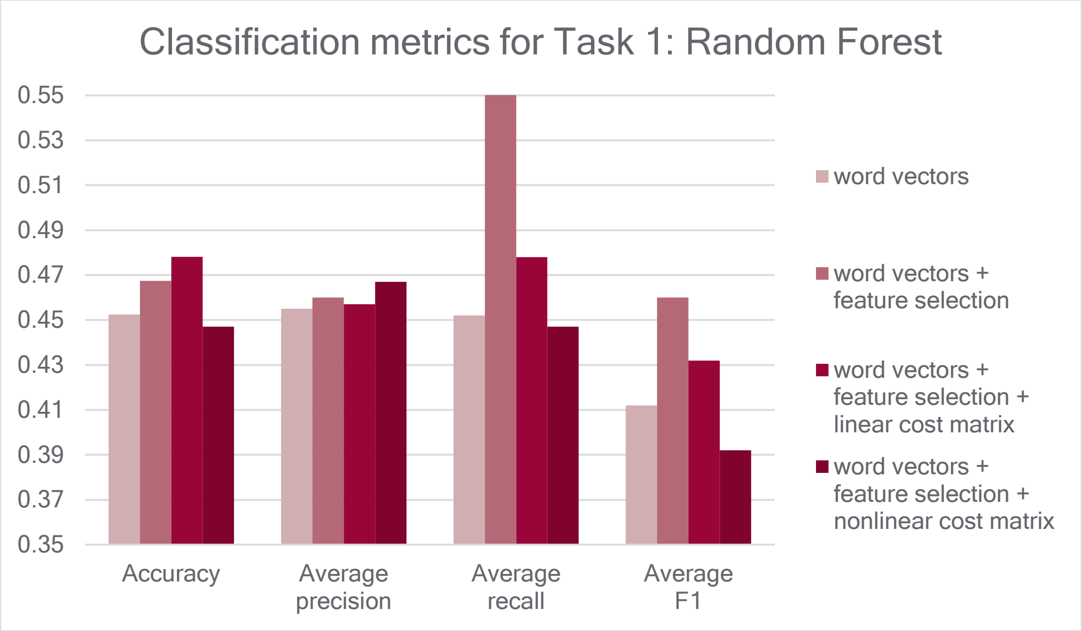


Figure 2: Classification results of Task 1 by J48 algorithm

To see more details into classification, the confusion matrices are reported below, whose structure is quite different between the two algorithms

Confusion matrix with highest accuracy by J48 (word vectors + feature selection + linear cost matrix) on the testing dataset

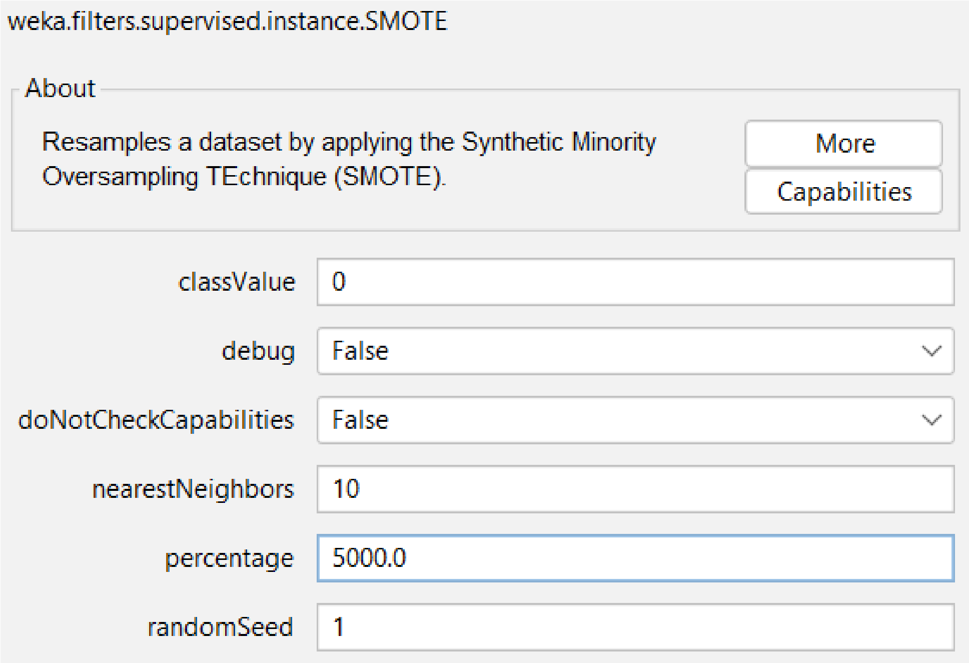
Actual ↓ /Predicted →	**1 star**	**2 star**	**3 star**	**4 star**	**5 star**
1 star	27	7	12	13	14
2 star	15	14	6	18	14
3 star	11	10	31	65	39
4 star	12	24	48	136	108
5 star	11	10	10	106	174

Confusion matrix with highest accuracy by Random Forest (word vectors + feature selection + linear cost matrix) on the testing dataset

Actual ↓ /Predicted →	**1 star**	**2 star**	**3 star**	**4 star**	**5 star**
1 star	38	2	3	20	10
2 star	12	3	4	32	16
3 star	2	0	10	94	50
4 star	5	1	12	165	145
5 star	1	0	4	75	231

4.2 Task 2 results

After converting the title and the text to word vectors, we can proceed to apply SMOTE to oversample the influential reviews by 50 times, which helps rebalance the dataset from 282 ratio of uninfluential/influential to 6.



weka.filters.supervised.instance.SMOTE

About

Resamples a dataset by applying the Synthetic Minority Oversampling TEchnique (SMOTE).

classValue 0

debug False

doNotCheckCapabilities False

nearestNeighbors 10

percentage 5000.0

randomSeed 1

Due to the unbalanced dataset, accuracy and precision are quite relevant since we only focus on catching influential reviews. Therefore, we would only need to compare the true positive rate, recall, ROC area and PRC area for class 1.

- True positive rate: it measures the proportion of actual positives (influential reviews) that are correctly identified by the model

- Recall: High recall means that the model is effective at detecting all relevant influential reviews when it is indeed influential

- ROC area: A larger area under the curve (AUC) indicates better model performance, with 1.0 means a perfect test and 0.5 means no discriminative ability by the model.

- Unlike the ROC curve, the precision-recall curve (PRC) focuses on the performance of positive class, which is useful for imbalanced dataset.

After testing the four cases, the classification can be reported for J48 and Random Forest across TP rate, recall, ROC area and PRC area. For each score, we could compare side by side the four study cases

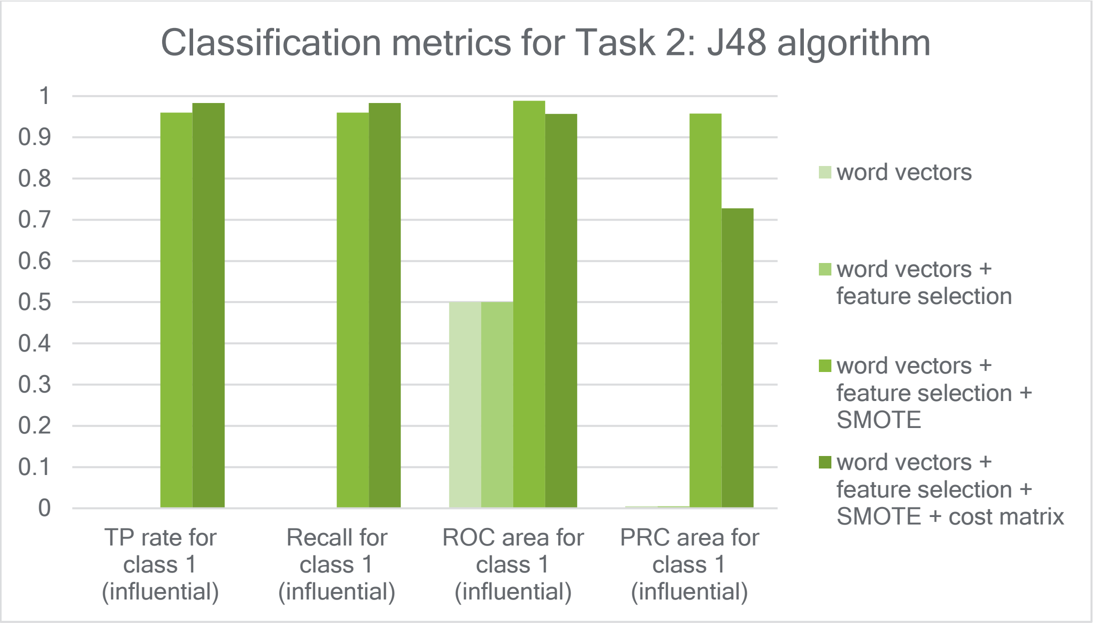


Figure 3: Classification results of Task 2 by J48 algorithm

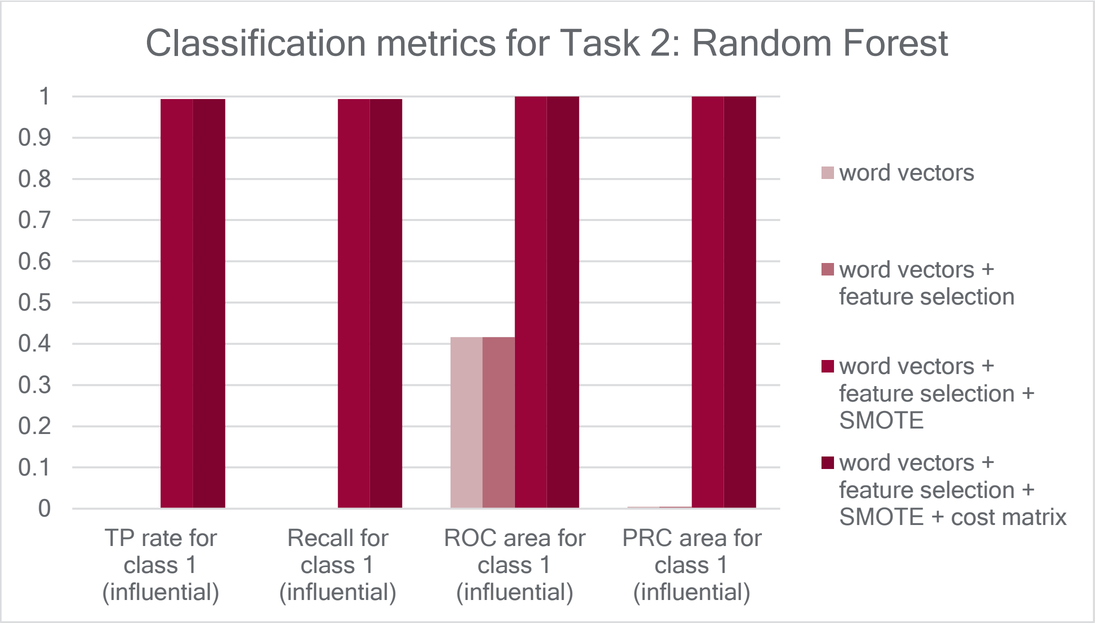


Figure 4: Classification results of Task 2 by Random Forest

Finally, the confusion matrices are reported for the best performing models.

Confusion matrix with highest recall by J48 (word vectors + feature selection + SMOTE) on the testing dataset

Actual ↓ /Predicted →	**Uninfluential review (0)**	**Influential review (1)**
Uninfluential review (0)	921	3
Influential review (1)	7	169

Confusion matrix with highest recall by Random Forest (word vectors + feature selection + SMOTE) on the testing dataset

Actual ↓ /Predicted →	**Uninfluential review (0)**	**Influential review (1)**
Uninfluential review (0)	924	0
Influential review (1)	1	175

5. Discussion

5.1 Task 1 results analysis

- Random forest tends to perform better than J48 on all metrics. This is expected because random forest is very resistant against overfitting, while J48 only have one single tree, which can easily overfit the data.
- Feature selection does indeed significantly improve the classification results. By removing irrelevant words, the classifiers would be less prone to overfitting on noise and can better generalize over testing data.
- The introduction of cost sensitive analysis further improves the classification score, suggesting that it would be more beneficial to wrongly classify hotel ratings off by 1 or 2 stars rather than 3 or 4 stars in the extreme case. Cost sensitive analysis is highly important in business settings since predicting hotel ratings with seriously wrong ratings would reduce the reputation of hospitability companies like TripAdvisor.
- Linear cost matrix results in better performance for two classifiers J48 and Random Forest than nonlinear cost matrix, so we could assume that the real life cost of wrong rating prediction increases linearly by each star.

Based on the confusion matrices of J48 and Random Forest in Task 1, we can make some mild assumptions as follows

1. It is quite easy to truly predict the rating as high stars (4 or 5 stars) if the review praises the hotel. However, the converse should not be true, as the textual content should not always be dismissing in the reviews of lower rating stars (1 to 3 stars). That is why for lower stars, J48 and Random Forest tend to mistake them for higher stars. However, if the review is indeed truly of higher stars, the classifiers are highly confident that it has high star ratings.
2. Out of 5 stars, it appears that 5 stars is the easiest to predict, and next is 4 stars. Then, 1 star is a little easier to predict since the connotation of bad criticism is very heavy and sometimes harsh. The hardest star to predict appears to be 2 stars and 3 stars, since they are lying in the grey zone, as the customers could be praising or criticizing. As a result, the classifiers have a difficult problem in determining their true stars.

5.2 Task 2 results analysis

Because there are only 11 influential reviews in the whole dataset, it is helpful to investigate one influential review to judge how it feels like. One of them is

Title: Great Hotel - Worries about complaints totally unfounded
Review text: Shortly before travelling to NY I looked on this website for updated reviews for the Edison Hotel. I was horrified to see people complaining about the smells, cockroaches in the rooms, lack of towels, ramshackle furniture etc. As I had booked this break 10 months earlier it was going to cost me an additional £750 to change hotels. To say the least I was not looking forward to this part of our tour. I was therefore pleasantly surprised when we arrived on 17th September for 4 nights. Apart from the great location, the room was plenty big enough, the bathroom although small was adequate and while it was old it was spotlessly clean with masses of hot water. Towels (yet another complaint) were changed daily and were plentiful. Additional pillows were provided within minutes of our request. The staff were helpful and I particularly liked it when the bell captain told the taxi driver how much we should expect to be paying so there was no fear of being ‘ripped off’. At one end of the street you could pick up the Greyline coach for the Uptown Loops and at the other end of the street you

could pick up the coach for the other Loops. The diner attached to the hotel was excellent value and a satisfying breakfast of eggs, toast and coffee could be had for around $5. Would definitely stay here again.

Based on this review, we could make a few assumptions about influential review

1. The reviews tend to be very detailed in describing the stay, including specifics about room conditions, amenities, hotel services, and even interactions with the staff.

2. Many influential reviews mention both positives and negatives of their stay with the hotel. This balanced approach makes a review seem more credible and trustworthy

3. Many influential reviews mention the convenience of the location of the accommodation, which can be highly valuable to travelers

4. Emotional language expressing satisfaction or disappointment from the reviewer can create a stronger decisive response from readers.

Regarding the classification result, the baseline word vectors, and the feature selection versions fail to detect any influential reviews, which is extremely difficult since influential reviews are so rare, so their detection is extremely unlikely. As a result, we should resort to SMOTE to balance out the two classes in hope of better results. It is indeed true that after applying SMOTE techniques, the classifier performs much better and no longer have nonzero recall.

From the confusion matrices, while it is uncertain if J48 has indeed managed to classify any true samples of influential review as influential, since there are 10 data misclassified. However, for Random Forest, it is certain that after applying SMOTE technique, it has managed to classify all true influential reviews as influential. Therefore, we can conclude that Random Forest performs better than J48. Finally, the cost matrix, on the contrary to initial assumption, does not improve the performance for this second task at all.

6. Conclusion

The analysis of classification tasks in this report has shown assumptions, results and analysis for two classification tasks of hotel star ratings and influential review detection. To recap, in Task 1, Random Forest consistently outperformed J48, showing that it is robust against overfitting, and it is better equipped to handle large word vectors through ensemble learning. The application of feature selection significantly improved model performance by reducing noise and focusing on relevant tokens (or TF-IDF score). Moreover, the introduction of cost-sensitive matrix helps reflect the practical business case for accurate rating prediction.

In Task 2, the rarity of influential reviews posed makes it extremely difficult for detection. The application of SMOTE is critical to improve the model's sensitivity to the minority influential review. The success of Random Forest after applying SMOTE shows us that ensembled trees coupled with synthetic data generation can overcome the biases introduced by skewed data distribution.

Finally, this report is only a preliminary test to the classification. In order for the results to be useful, the classifiers should be heavily calibrated, and the word vector settings should be well chosen. When the models are successful in prediction, they can finally be deployed to extract meaningful details from the Facebook dataset.

References

[1] anaN, N.Sarav & thri, V.Gaya. (2018). Performance and Classification Evaluation of J48 Algorithm and Kendall’s Based J48 Algorithm (KNJ48). International Journal of Computer Trends and Technology. 59. 73-80. 10.14445/22312803/IJCTT-V59P112.

[2] Breiman, L. (2001). Random Forests. Machine Learning. 45. 5-32. 10.1023/A:1010950718922.

[3] Desai, Ankit & Jadav, Prashant. (2012). An Empirical Evaluation of Ad boost Extensions for Cost-Sensitive Classification. International Journal of Computer Applications. 44. 34-41. 10.5120/6325-8677.

[4] : T I Pehlivanova and V I Nedeva 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1031 012055

[5] Chawla, Nitesh & Bowyer, Kevin & Hall, Lawrence & Kegelmeyer, W.. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. 10.1613/jair.953.