Business Report

Predicting Review Author Influence

Real Life vs. Online Environment

CONTENTS

1. INTRODUCTION

This report presents an analysis of the review author's influence using machine learning techniques, specifically focusing on Tasks 3 and 4 outlined in the provided dataset.

- Task 3: predicting the real-life influence of review authors based on their travel history (Author_num_cities)
- Task 4: predicting the online influence of review authors based on their review helpfulness votes (Author_num_helpful_votes)

Problem statement: Identifying Influential Review Authors (in real life and online environment).

Objective:

The objective is to develop accurate machine learning models capable of predicting review author influence, aiding in targeted marketing efforts, customer engagement strategies, and business decision-making.

Motivation and Business Value:

Identifying influential review authors is crucial for tailoring promotional activities, enhancing customer engagement, and maximizing business impact. By accurately predicting influential authors, the company can optimize resources, improve customer satisfaction, and drive revenue growth.

Target Audience:

Insights derived from the analysis can be utilized by teams directly involved in customer acquisition, retention, and engagement within the organization, including:

- Marketing Team: to tailor targeted campaigns (especially for highly influential review authors), optimize advertising strategies, allocate resources effectively, maximize brand visibility and attract new guests.
- Customer Service Team: to enhance personalized communication, prioritize interactions with high-influential customers, ensure personalized and exceptional experiences to enhance satisfaction and encourage positive reviews.
- Operations Team: to tailor amenities, services, and facilities to meet the preferences and expectations of influential guests, thereby enhancing overall guest satisfaction and loyalty.
- Revenue Management Team/Hotel owners: to guide revenue management strategies, allowing for dynamic pricing, package offerings, and inventory allocation to capitalize on the preferences and booking behaviors of highly influential customers.

2. PREDICTIVE MODEL FOR REVIEW AUTHOR INFLUENCE: REAL LIFE

2.1 DATA PROCESSING

First, by using "Add Expression" filter in Weka with expression = ifelse ('Author_num_cities' > 15, 1, 0), we can add extra column as binary variable to check whether the author has visited more than 15 cities ('Author_num_cities' > 15):

- Class 0 includes the author who has visited less than or equal to 15 cities.
- Class 1 includes the author who has visited more than 15 cities.

Then, this newly created column is converted to nominal variables, using "NumericaltoNominal" filter in Weka. Also, the author's location is converted to nominal format using "StringtoNominal" filter.

A cost matrix is added to penalize misclassifications, particularly false negatives where authors who have actually visited more than 15 cities are erroneously classified as not having done so. By doing this, the company can:

- Focus on Target Group: By penalizing false negatives, the model prioritizes correctly identifying individuals who have visited more than 15 cities. This aligns with the company's specific interest in this subgroup of reviewers, potentially for targeted marketing efforts or other initiatives, which will be discussed in Chapter 4.
- Mitigate Losses: Misclassifying influential reviewers as non-influential (false negatives) could result in missed opportunities for the company, such as failing to engage with potential brand advocates or missing valuable feedback. Hence, penalizing these misclassifications helps mitigate potential losses associated with overlooking influential individuals.

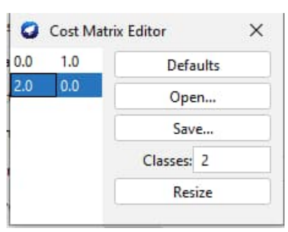


Figure 1: Cost Matrix

2.2 MODEL TRAINING & TESTING

The dataset is split into train and test sets with a test size of 20%:

- Training dataset: 80% (2494 instances)
- Test dataset: 20% (624 instances)

Now we can have a quick glance at the training dataset. There is an imbalance in the dataset: 468 (18.28% - red color) of authors visiting more than 15 cities and 2038 (81.72% - blue color) of authors not.

The goal is to predict the minority class (authors visiting more than 15 cities).

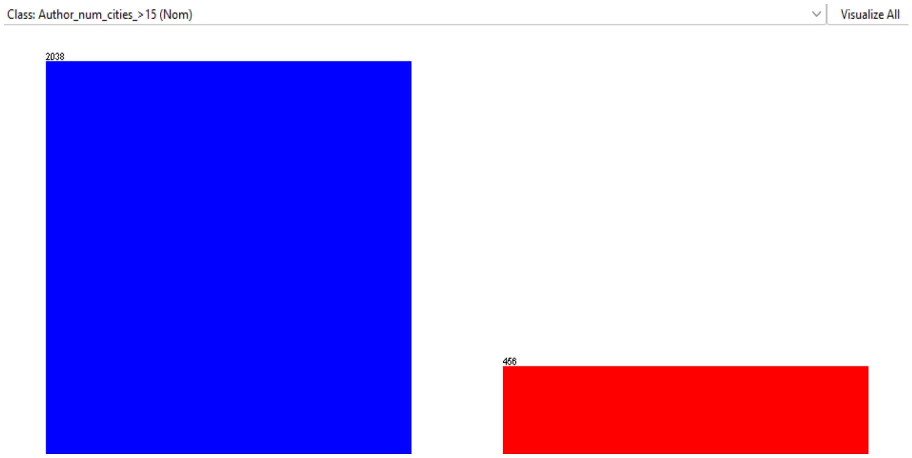


Figure 2: Original Training Dataset overview - Task 3

The models chosen to be trained are OneR, Decision Tree J48, Random Forest, Cost Sensitive Classification (J48/Random Forest/OneR as the base classifier) using the original imbalanced data and rebalanced data using SMOTE methods. The reasoning for using the above-mentioned models is:

Classifier	Advantage
OneR	Simple yet powerful rule-based algorithm – use one single predictor variable to predict a dependent variable (Author_num_cities >15)
J48	Suitable for handling categorical data. Provide a good interpretability to the prediction. Easy to visualize
Random Forest	Avoid overfitting. Enhance predictive accuracy and generalization performance
Cost Sensitive Classification	Prioritize the correct identification of the target class (review authors who visited more than 15 cities. Mitigate the impact of misclassification costs. Suitable for imbalanced dataset

Regarding the process of training models, the models' accuracy performance is first compared using Weka Experimenter (Cross-validation method). Then, we ran the test again using "percentage split" (75%) and finally tested the trained models with the supplied test set.

We conducted training on models through 3 rounds.

ROUND 1 – NO SMOTE, INCLUDE ALL VARIABLES

In the first round, we utilize imbalanced training dataset and all available variables.

- <u>Attribute selection:</u> 16 attributes (15 attributes are predictor variables. The last attribute “Author_num_cities_>15” is the class/dependent variable).

```
Attributes:   16
              via_mobile
              revisit
              Rating_overall
              Rating_service
              Rating_cleanliness
              Rating_value
              Rating_location
              Rating_sleep_quality
              Rating_rooms
              Rating_check_in_front_desk
              Rating_business_service_(e_g_internet_access)
              Author_num_helpful_votes
              Author_num_reviews
              Author_location
              Author_num_helpful_votes >100
              Author_num_cities_>15
```

- <u>Model evaluation:</u>

First, we want to compare the accuracy of different models by using Weka Experimenter. The testing method used here is cross-validation. Based on this result, OneR model outperformed other models. The same result can be achieved using “percentage split” testing method.

```
Analysing:  Percent_correct
Datasets:   1
Resultsets: 7
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       4/2/24, 3:50 PM


Dataset                  (1) rules.Ze | (2) trees (3) trees (4) meta. (5) meta. (6) meta. (7) rules
----------------------------------------------------------------------------------------------------
'R_data_frame-weka.filter(100)   81.72 |   92.89 v   90.88 v   92.43 v   93.14 v   92.69 v   93.41 v
----------------------------------------------------------------------------------------------------
                               (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(4) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.J48 -- -C 0.25 -M 2' -110658209263002404
(5) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -110658209263002404
(6) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W rules.OneR -- -B 6' -110658209263002404
(7) rules.OneR '-B 6' -3459427003147861443
```

However, given the imbalanced nature of the dataset and the goal of predicting the minority class (authors visiting more than 15 cities), it is important to consider models that effectively handle class imbalance and prioritize the correct identification of the minority class (Class 1).

Hence, besides total cost of each model, we focus on its Recall rate of Class 1.

Good ROC area performance of models with unbalanced data is explained by the fact that they have high accuracy in predicting the non-influential review authors in real life i.e., True Negatives (authors visiting less than 15 cities) – which is not our focus.

Classifiers	Evaluation metrics (based on supplied test set)					
	Accuracy	Precision of Class 1	Recall of Class 1	F-measure of Class 1	ROC Area	Total Cost
OneR	95.19%	0.879	0.813	0.845	0.911	50
Cost sensitive (OneR)	95.03%	0.863	0.897	0.879	0.93	44
J48	94.87%	0.927	0.81	0.864	0.985	56
Cost sensitive (J48)	95.19%	0.907	0.849	0.877	0.962	49
Random Forest	92.63%	0.955	0.667	0.785	0.979	88
Cost sensitive (Random Forest)	94.23%	0.869	0.841	0.855	0.973	56

Table 1: Results of classification models - Task 3 Round 1

Based on the above results, we can see that OneR is no longer the optimal model, due to low Recall rate of Class 1, indicating this model does not perform so well in predicting True Positive cases of the minority class.

- Model selection:

Considering the goal of predicting the minority class while balancing precision and recall, we selected the final models, which have highest recall of class 1 while keeping the overall cost of misclassifications relatively low. Cost Sensitive (OneR) stands out. It achieves high recall of Class 1, indicating a better ability to identify actual instances of the minority class, along with a lower Total Cost compared to other models.

Therefore, Cost Sensitive (OneR) would be recommended for predicting authors visiting more than 15 cities in this imbalanced dataset.

```
Classifier Model
Author_num_reviews:
        < 24.5  -> 0
        < 25.5  -> 1
        < 26.5  -> 0
        < 33.5  -> 1
        < 34.5  -> 0
        >= 34.5 -> 1
(2343/2494 instances correct)


Cost Matrix
 0 1
 2 0


Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances         593               95.0321 %
Incorrectly Classified Instances        31                4.9679 %
Kappa statistic                          0.8481
Total Cost                              44
Average Cost                             0.0705
Mean absolute error                      0.0497
Root mean squared error                  0.2229
Relative absolute error                 15.9703 %
Root relative squared error             55.4621 %
Total Number of Instances              624

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.964    0.103    0.974      0.964    0.969      0.848    0.930     0.967     0
                 0.897    0.036    0.863      0.897    0.879      0.848    0.930     0.794     1
Weighted Avg.    0.950    0.090    0.951      0.950    0.951      0.848    0.930     0.932

=== Confusion Matrix ===

   a   b   <-- classified as
 480  18 |   a = 0
  13 113 |   b = 1
```

Figure 3: Cost Sensitive Classifier (OneR as the base dataset) – imbalanced dataset.

ROUND 2 – SMOTE, INCLUDE ALL VARIABLES

Now, we are interested in balancing training dataset so that the model is not biased towards the majority class (Class 0, where authors visit less than 15 countries). Hence, in this round, we applied SMOTE method (applying default setting) to increase instances of Class 1.

After using SMOTE, total instances of training data set increased to 2950.

- <u>Attribute selection:</u> Same as above - 16 attributes (15 attributes are predictor variables. The last attribute "Author_num_cities_>15" is the class)

```
Attributes:   16
              via_mobile
              revisit
              Rating_overall
              Rating_service
              Rating_cleanliness
              Rating_value
              Rating_location
              Rating_sleep_quality
              Rating_rooms
              Rating_check_in_front_desk
              Rating_business_service_(e_g_internet_access)
              Author_num_helpful_votes
              Author_num_reviews
              Author_location
              Author_num_helpful_votes >100
              Author_num_cities_>15
```

- <u>Model evaluation:</u>

Based on accuracy performance (cross-validation testing), Random Forest stands out to be the best model.

```
Dataset                   (1) rules.Ze | (2) trees (3) trees (4) meta. (5) meta. (6) meta. (7) rules
----------------------------------------------------------------------------------------------------
'R_data_frame-weka.filter(100)   69.08 |   92.01 v   95.03 v   91.34 v   94.65 v   93.18 v   92.90 v
----------------------------------------------------------------------------------------------------
                              (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(4) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.J48 -- -C 0.25 -M 2' -110658209263002404
(5) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -110658209263002404
(6) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W rules.OneR -- -B 6' -110658209263002404
(7) rules.OneR '-B 6' -3459427003147861443
```

However, we prioritize the correct identification of the minority class (Class 1). Hence, same as above, we compared Recall rate of Class 1 in these models (based on the supplied test set). The result is presented below:

Classifiers	Evaluation metrics (based on supplied test set)					
	Accuracy	Precision of Class 1	Recall of Class 1	F-measure of Class 1	ROC Area	Total Cost
OneR	94.71%	0.85	0.897	0.873	0.928	46
Cost sensitive (OneR)	94.39%	0.823	0.921	0.869	0.935	45
J48	92.48%	0.745	0.952	0.836	0.935	53
Cost sensitive (J48)	90.55%	0.683	0.992	0.809	0.941	60
Random Forest	93.75%	0.86	0.825	0.842	0.975	61
Cost sensitive (Random Forest)	93.27%	0.792	0.905	0.844	0.976	54

Table 2: Results of classification models - Task 3 Round 2

In general, we can see clearly that every model's ability to predict True Positives (Recall rate) of Class 1 increases significantly with balanced data. Meanwhile, despite having the highest accuracy, Random Forest is no longer the optimal model due to extreme high cost (indicating that this model wrongly classifies the minority class as we set in the cost matrix).

J48 and Cost Sensitive classifier (using J48 as the base classifier) have the highest recall rate of Class 1 but also have the lowest precision rate, indicating that these classifiers generate a significant number of False Positive predictions. In other words, there are instances where the classifier incorrectly identifies review authors as highly influential in real life (authors visiting more than 15 cities) when they are actually not. Hence, these models do not perform well.

- <u>Model Selection:</u>

Considering the goal of predicting Class 1 with the highest recall rate and lowest total cost, the Cost sensitive (OneR) model appears to be the most suitable choice. It achieves a relatively high recall rate of Class 1 (0.921) while maintaining a lower total cost (45) compared to other models with similar recall rates. Therefore, Cost sensitive (OneR) strikes a good balance between effectively capturing instances of Class 1 and minimizing the total cost associated with misclassifications.

```
Author_num_reviews:
        < 19.090498     -> 0
        < 19.961036     -> 1
        < 20.009418500000002    -> 0
        < 20.842925     -> 1
        < 21.117804     -> 0
        < 21.9685305    -> 1
        < 22.015681999999998    -> 0
        < 23.845332499999998    -> 1
        < 24.0504055    -> 0
        >= 24.0504055   -> 1
(2815/2950 instances correct)


Cost Matrix
 0 1
 2 0


Time taken to build model: 0.04 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances         589               94.391  %
Incorrectly Classified Instances        35                5.609  %
Kappa statistic                          0.8334
Total Cost                              45
Average Cost                             0.0721
Mean absolute error                      0.0561
Root mean squared error                  0.2368
Relative absolute error                 14.5196 %
Root relative squared error             56.9935 %
Total Number of Instances              624

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.950    0.079    0.979      0.950    0.964      0.836    0.935     0.970     0
                 0.921    0.050    0.823      0.921    0.869      0.836    0.935     0.773     1
Weighted Avg.    0.944    0.073    0.948      0.944    0.945      0.836    0.935     0.930

=== Confusion Matrix ===

   a   b   <-- classified as
 473  25 |   a = 0
  10 116 |   b = 1
```

Figure 4: Cost sensitive classifier (OneR as the base classifier) - balanced dataset.

The base classifier OneR in Cost Sensitive classifier predicted authors' real-life influence (whether authors visit more than 15 cities) solely based on the number of reviews that the review authors have produced at TripAdvisor (Author_num_reviews). This model is not useful in practice, since it is fully based Author_num_reviews - the number of reviews that the review authors have produced at TripAdvisor. This is quite a common practice, when the author writes more reviews, he/she tends to be more likely to visit more than 15 cities.

Hence, for the next round, we aim to build a predictive model based on other attributes and exclude number of authors' reviews (Author_num_reviews) from the analysis.

ROUND 3 – SMOTE, ATTRIBUTE SELECTION (EXCLUD. AUTHOR_NUM_REVIEWS)

- <u>Attribute selection:</u>

In this round, we excluded Author_num_reviews attribute and selected remaining attributes based on their info gain worth by using "Attribute selection" filter with parameters E"InfoGainAttributeEval" – S"Ranker – T0.0-N-1" (Info Gain → Ranker → threshold = 0). After running the filters, 10 selected predictor attributes are:

No.	
1	Author_location
2	Author_num_helpful_votes
3	Author_num_helpful_votes >100
4	Rating_value
5	Rating_overall
6	revisit
7	Rating_rooms
8	Rating_service
9	Rating_cleanliness
10	via_mobile
11	Author_num_cities_>15

The model is trained using balanced training dataset (SMOTE method as utilized in Round 2).

- <u>Model evaluation:</u>

Based on accuracy performance (cross-validation testing), Random Forest stands out to be the best model.

```
Dataset                   (1) rules.Ze | (2) trees (3) trees (4) meta. (5) meta. (6) meta.
------------------------------------------------------------------------------------------
'R_data_frame-weka.filter(100)   69.08 |   84.06 v   89.76 v   82.52 v   87.14 v   78.74 v
------------------------------------------------------------------------------------------
                               (v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428698
(4) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.J48 -- -C 0.25 -M 2' -110658209263002404
(5) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W trees.RandomForest -- -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -110658209263002404
(6) meta.CostSensitiveClassifier '-cost-matrix \"[0.0 1.0; 2.0 0.0]\" -S 1 -W rules.OneR -- -B 6' -110658209263002404
```

Nevertheless, our primary focus lies in accurately identifying the minority class (Class 1). Therefore, we assess the Recall rate of Class 1 across these models using the provided test set. The outcome is detailed below:

Classifiers	Evaluation metrics (based on supplied test set)					
	Accuracy	Precision of Class 1	Recall of Class 1	F-measure of Class 1	ROC Area	Total Cost
OneR	80.77%	0.542	0.31	0.394	0.622	207
Cost sensitive (OneR)	74.36%	0.365	0.365	0.365	0.602	240
J48	86.86%	0.631	0.841	0.721	0.883	125
Cost sensitive (J48)	81.90%	0.53	0.913	0.671	0.854	124
Random Forest	88.62%	0.816	0.563	0.667	0.905	126

Cost sensitive (Random Forest)	85.90%	0.623	0.762	0.686	0.905	118

Table 3: Results of classification models - Task 3 Round 3

As we can see, after removing Author_num_reviews, there are strong variation in ROC area of all models. OneR and Cost Sensitive(OneR) exhibit extremely low Recall rate of Class 1 with high cost, indicating that the models can predict only low proportion of True Positive instances (Class 1) out of all actual positive instances.

Meanwhile, in terms of Recall rate of Class 1 and total cost, J48, Cost sensitive (J48), Random Forest and Cost Sensitive (Random Forest) are outperforming.

- J48 and Cost sensitive (J48): Both models have comparatively high Recall rate of Class 1 and similar cost. However, Cost sensitive (J48) has extremely lower precision rate of Class 1 and overall accuracy. Precision rate is 0.53, indicating that nearly half of the instances predicted as positive may be false positives and the model's performance is just as bad as random guessing. Hence, J48 is outperforming cost sensitive (J48).
- Random Forest and Cost sensitive (Random Forest): Cost sensitive (Random Forest) achieves higher recall rate of Class 1 while keeping total cost lower than Random Forest.

Hence, after being short-listed, J48 and Cost sensitive (Random Forest) are two outstanding models.

- Model selection:

Between these 2 models, Cost sensitive (Random Forest) model is recommended as the best option for predicting authors with high real-life influence (Class 1) due to the following reasons:

- Despite having slightly lower Recall rate of Class 1, Cost sensitive (using Random Forest) exhibit higher ROC area and lower cost (118 compared to 125):
  - ROC area of 0.905 suggests strong discrimination ability.
  - Importantly, the total cost associated with this model is relatively low at 118, demonstrating cost-effectiveness in terms of misclassification errors.
- Random Forest tends to be more robust against overfitting compared to decision trees like J48, as it averages multiple decision trees. The above result is achieved when we tested with one supplied test set. Hence, for generalization in future, this robustness is needed, which can lead to more stable performance, especially when dealing with noisy or complex datasets.

```
Time taken to build model: 2.43 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.11 seconds

=== Summary ===

Correctly Classified Instances         536               85.8974 %
Incorrectly Classified Instances        88               14.1026 %
Kappa statistic                          0.596
Total Cost                             118
Average Cost                             0.1891
Mean absolute error                      0.2699
Root mean squared error                  0.3349
Relative absolute error                 69.866  %
Root relative squared error             80.5915 %
Total Number of Instances              624

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.238    0.936      0.884    0.909      0.601    0.905     0.966     0
                 0.762    0.116    0.623      0.762    0.686      0.601    0.905     0.777     1
Weighted Avg.    0.859    0.214    0.873      0.859    0.864      0.601    0.905     0.928

=== Confusion Matrix ===

   a   b   <-- classified as
 440  58 |   a = 0
  30  96 |   b = 1
```

Figure 5: Cost sensitive classifier (Random Forest as the base classifier)

2.3 CONCLUSION

To sum up, achieving better prediction performance involves two key strategies: balancing the training dataset and selecting appropriate models based on dataset attributes. Depending on the presence or absence of the "Author_num_reviews" attribute in the original dataset, different models are recommended:

- Scenario 1: Original Dataset Includes "Author_num_reviews" attribute:

✓ Preferred Model: Cost Sensitive (OneR)
✓ Justification: This model effectively balances precision and recall while maintaining cost-effectiveness, making it the preferred choice for such scenarios.

- Scenario 2: Original Dataset Excludes "Author_num_reviews" attribute:

✓ Attribute Selection: Utilize an attribute selection method based on their information gain to identify the most relevant predictor attributes.
✓ Preferred Model: Cost Sensitive (Random Forest)
✓ Justification: After selecting relevant attributes, this model offers strong discrimination ability and cost-effectiveness, ensuring reliable predictions even in the absence of the "Author_num_reviews" attribute.

3. PREDICTIVE MODEL FOR REVIEW AUTHOR INFLUENCE: ONLINE ENVIRONMENT

3.1 DATA PROCESSING

Data preprocessing followed a similar approach to the previous chapter's methodology:

1. Add a binary variable (called “'Author_num_helpful_votes' > 100”) to check whether authors received more than 100 helpfulness votes: Use the "Add Expression" filter in Weka with the expression ifelse('Author_num_helpful_votes' > 100, 1, 0). Class 0 encompasses authors with fewer than or equal to 100 helpfulness votes, while Class 1 comprises those with over 100 helpfulness votes.
2. Convert this binary variable to nominal variable via the "NumericaltoNominal" filter in Weka. Furthermore, the authors' locations were transformed into nominal format using the "StringtoNominal" filter.

A cost matrix was integrated to penalize misclassifications, with a particular focus on false negatives, where authors who have genuinely received over 100 helpfulness votes are incorrectly classified as not having done so. The main reason is that it is most important for the company to accurately identify high-influential authors so that appropriate measures can be implemented to leverage their influence. It is more costly if the true high-influential authors are overlooked compared to directing influence-utilizing actions towards some customers who may not be high-influential in online environments.

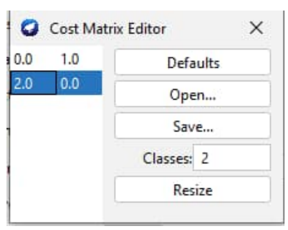


Figure 6: Cost matrix

3.2 MODEL TRAINING & TESTING

Similarly, the dataset is split into train and test sets with a test size of 20%:

- Training dataset: 80% (2494 instances)
- Test dataset: 20% (624 instances)

There is a heavy imbalance in the training dataset: 71 (2.85% - red color) authors receiving more than 100 helpfulness votes and 2423 (97.15% - blue color) authors not.

The goal is to predict the minority class (authors receiving more than 100 helpfulness votes).

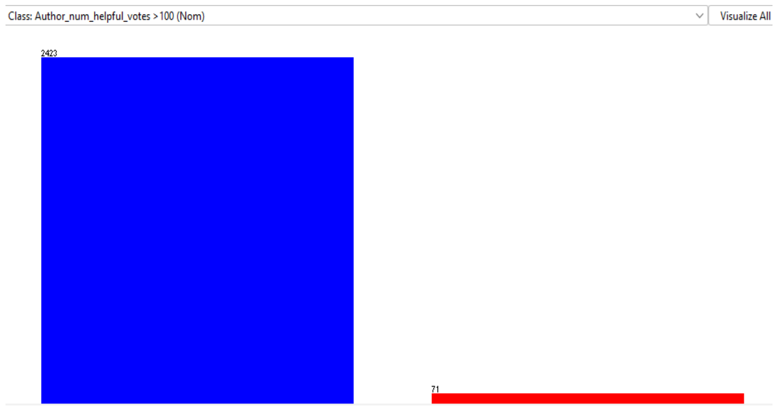


Figure 7: Original Training Dataset overview – Task 4

As learnt from the previous task, every model's ability to predict True Positives increases significantly with balanced training dataset. Hence, we used balanced training dataset by implementing SMOTE methods.

The models chosen to be trained are Cost Sensitive Classifier using different classifiers as the base classifiers, such as OneR, Decision Tree J48, Random Forest, SMO, Simple Logistics and Bagging. Cost-sensitive classification techniques are particularly suitable, as they allow the model to explicitly account for the imbalance between classes by assigning different misclassification costs, penalizing misclassifications differently and giving higher penalties to errors in the minority class.

The reasoning for using the above-mentioned classifiers as the base classifier:

Classifier	Advantage
OneR	As mentioned in Chapter 2
J48	As mentioned in Chapter 2
Random Forest	As mentioned in Chapter 2
SMO	Support Vector Machine (SVM) implementation suitable for handling non-linear decision boundaries and capturing complex relationships in the data.
Simple Logistics	Suitable for modeling linear relationships between predictors and the target variable (Author_num_helpful_votes > 15)
Bagging	Reduces variance by aggregating predictions from multiple models trained on bootstrap samples. It improves stability and robustness, particularly in high-variance models.

We conducted training on models through 2 rounds. As the goal is to correct predict class 1 (the minority class, including review authors with high influence in online environment), we focus on Recall rate of Class 1 and total misclassifications cost.

ROUND 1 – SMOTE, INCLUDE ALL VARIABLES

In this first round, we will include all available variables in our analysis.

- Attribute selection: 17 attributes (16 attributes are predictor variables. The last attribute "Author_num_helpful_votes_>100" is the class/dependent variable).

No.	
1	num_helpful_votes
2	via_mobile
3	revisit
4	Rating_overall
5	Rating_service
6	Rating_cleanliness
7	Rating_value
8	Rating_location
9	Rating_sleep_quality
10	Rating_rooms
11	Rating_check_in_front_desk
12	Rating_business_service_(e_g_internet_access)
13	Author_num_cities
14	Author_num_reviews
15	Author_location
16	Author_num_cities_>15
17	Author_num_helpful_votes_>100

- Model evaluation:

Classifiers	Evaluation metrics (based on supplied test set)						
	Accuracy	Precision of Class 1	Recall of Class 1	F-measure of Class 1	PRC Area of Class 1	ROC Area	Total Cost
Zero R	97.12%	-	-	-	-	0.5	36
Cost sensitive (OneR)	96.80%	0.458	0.611	0.524	0.291	0.795	27
Cost sensitive (J48)	93.43%	0.265	0.722	0.388	0.225	0.894	46
Cost sensitive (Random Forest)	97.44%	0.583	0.389	0.467	0.605	0.964	27
Cost sensitive (SMO)	97.77%	0.643	0.5	0.563	0.336	0.746	23
Cost sensitive (Simple Logistics)	97.92%	0.647	0.611	0.629	0.619	0.959	20
Cost sensitive (Bagging)	96.96%	0.462	0.333	0.387	0.247	0.695	31

Table 4: Results of classification models - Task 4 Round 1

ZeroR serves as a baseline model. In general, the majority of models underperform, where precision and recall rates of Class 1 are usually lower than 0.5 (highlighted as red), indicating that these models struggle to accurately identify highly influential authors. Despite having good accuracy, most models have recall rates lower than 0.5 and fail to capture a significant portion of actual highly influential authors (Class 1), which is our main focus.

- <u>Model Selection:</u>

Based on the analysis, the Cost Sensitive (Simple Logistics) model appears to be the best choice for predicting Class 1. It achieves balanced precision and recall for Class 1, with the lowest total cost among the models. Additionally, it has high PRC Area and ROC Area, indicating good discrimination ability and overall performance. Therefore, Cost Sensitive (Simple Logistics) provides the best balance between predictive performance and cost-effectiveness for predicting Class 1 in this scenario.

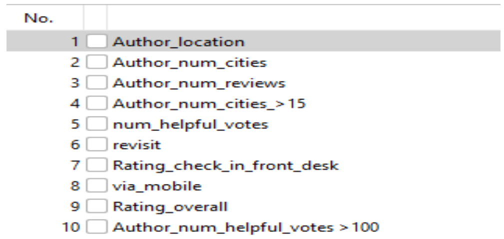
```
Cost Matrix
 0 1
 2 0


Time taken to build model: 21.44 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.03 seconds

=== Summary ===

Correctly Classified Instances         611               97.9167 %
Incorrectly Classified Instances        13                2.0833 %
Kappa statistic                          0.6179
Total Cost                              20
Average Cost                             0.0321
Mean absolute error                      0.0301
Root mean squared error                  0.1337
Relative absolute error                 37.0018 %
Root relative squared error             78.8676 %
Total Number of Instances              624

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.990    0.389    0.988      0.990    0.989      0.618    0.959     0.999     0
                 0.611    0.010    0.647      0.611    0.629      0.618    0.959     0.619     1
Weighted Avg.    0.979    0.378    0.979      0.979    0.979      0.618    0.959     0.988

=== Confusion Matrix ===

   a   b   <-- classified as
 600   6 |   a = 0
   7  11 |   b = 1
```

Figure 8: Cost Sensitive classifier (Simple Logistics as the base classifier)

ROUND 2 - SMOTE, ATTRIBUTE SELECTION

- <u>Attribute selection:</u>

For better tuning, in this round, we select attributes based on their info gain worth in predicting the class variable (Author_num_helpful_vote_>15), by using “Attribute selection” filter with parameters E”InfoGainAttributeEval” – S”Ranker – T0.0-N-1” (Info Gain → Ranker → threshold = 0). After running the filters, 9 selected predictor attributes are:



No.	
1	Author_location
2	Author_num_cities
3	Author_num_reviews
4	Author_num_cities_>15
5	num_helpful_votes
6	revisit
7	Rating_check_in_front_desk
8	via_mobile
9	Rating_overall
10	Author_num_helpful_votes >100

Now we are interested to see if the performance of these models has improved.

- <u>Model evaluation:</u>

Classifiers	Evaluation metrics (based on supplied test set)						
	Accuracy	Precision of Class 1	Recall of Class 1	F-measure of Class 1	PRC Area of Class 1	ROC Area	Total Cost
Zero R	97.12%	-	-	-	-	0.5	36
Cost sensitive (OneR)	95.83%	0.375	0.667	0.48	0.26	0.817	26
Cost sensitive (J48)	94.07%	0.289	0.722	0.413	0.236	0.852	42
Cost sensitive (Random Forest)	97.60%	0.615	0.444	0.516	0.56	0.921	25
Cost sensitive (SMO)	97.60%	0.615	0.444	0.516	0.29	0.718	25
Cost sensitive (Simple Logistics)	97.92%	0.632	0.667	0.649	0.611	0.958	19
Cost sensitive (Bagging)	97.12%	0.5	0.278	0.357	0.357	0.703	31

Table 5: Results of classification models - Task 4 Round 2

In general, recall rate of Class 1 are improved in Cost sensitive classifiers using OneR/Random Forest/Simple Logistics as the base classifier. Also, the total costs of each model are reduced.

- <u>Model Selection:</u>

Based on the provided evaluation metrics, the best model for predicting Class 1 appears to be Cost Sensitive (Simple Logistics). Besides having consistently high performance, in this round, this model exhibits increased recall rate of class 1 (from 0.611 to 0.667), which is among the highest values compared to other models. It also exhibits relatively low total cost (decrease from 20 to 19) compared to other models, which suggests cost-effectiveness in terms of misclassification errors.

3.3 CONCLUSION

In short, to achieve the best predictive performance of high-influential review authors in online environments (who have received more than 100 helpfulness votes), we advocate the following strategies:

- ✓ Data processing: balancing training dataset (using SMOTE method)
- ✓ Attribute selection: based on their information gain significance (using “Attribute selection” filter)
- ✓ Preferred model: Cost Sensitive classifier (using Simple Logistics as the base classifier)
- ✓ Justification: The model offers a good balance of performance metrics, including precision, recall, accuracy, and cost-effectiveness.

4 BUSINESS SUGGESTIONS

After identifying highly influential review authors across both online platforms and real-world interactions, the company can leverage this valuable to elevate its operational efficiency and strategic initiatives. Here are three refined strategies to maximize the impact:

1. Engagement and Recognition Programs:

- Develop customized engagement initiatives aimed at fostering stronger connections with highly influential authors pinpointed through the predictive model.
- Offer VIP experiences during their hotel stays, including complimentary room upgrades, personalized amenities, and exclusive access to premium facilities.
- Encourage influential authors to share their positive experiences on platforms like TripAdvisor, while prominently highlighting their reviews on the hotel's website and social media channels.
- Publicly acknowledge their contributions to incentivize continued feedback and engagement.

2. Influencer Collaboration and Partnerships:

- Forge collaborative partnerships with highly influential authors to amplify their impact and extend the hotel's reach.
- Co-create compelling marketing content, such as destination guides and experiential videos, highlighting the hotel's unique offerings.
- Leverage influencers' credibility to target specific demographics or niche markets aligned with the hotel's customer base.
- Execute targeted advertising campaigns on social media platforms, leveraging influencer content to drive engagement and bookings among their followers.

3. Guest Loyalty and Referral Programs:

- Launch a tailored loyalty program catering to highly influential reviewers, featuring exclusive benefits, discounts, and rewards for repeat stays and referrals.
- Encourage satisfied guests, particularly influential reviewers, to refer friends, family, and followers to the hotel through a structured referral program.
- Monitor referral metrics closely to gauge program effectiveness in driving new bookings and fostering guest loyalty.

By implementing these refined strategies, the company can harness the influence of high-profile reviewers to enhance guest experiences, boost brand visibility, and drive sustained business growth.