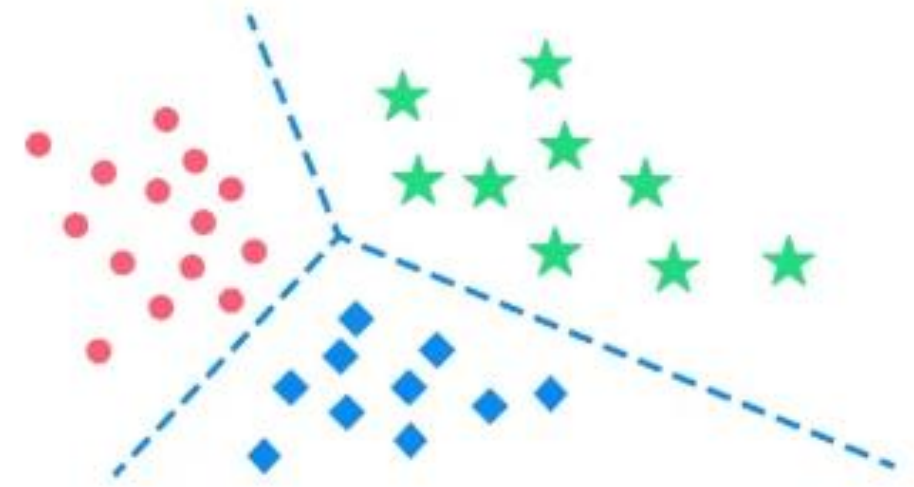
Customer feedback analysis

57E00500 - Capstone: Business Intelligence

Content of the analysis

This report presents results of an analysis with 4 topics of interest:

1. **Predicting hotels' overall rating** using textual feedback
2. **Discovering influential reviews** based on the textual feedback
3. **Discovering real-life influential authors** based on their full feedback
4. **Discovering online influential authors** based on their full feedback



Dataset

- The dataset used throughout this analysis contains hotel reviews from TripAdvisor
- The dataset contains 3118 reviews with 22 attributes
- The attributes can be grouped to
  - Author specific attributes (7)
  - Visit specific attributes (13)
  - Hotel id (1)
  - Number of helpfulness votes (1)

Data preprocessing

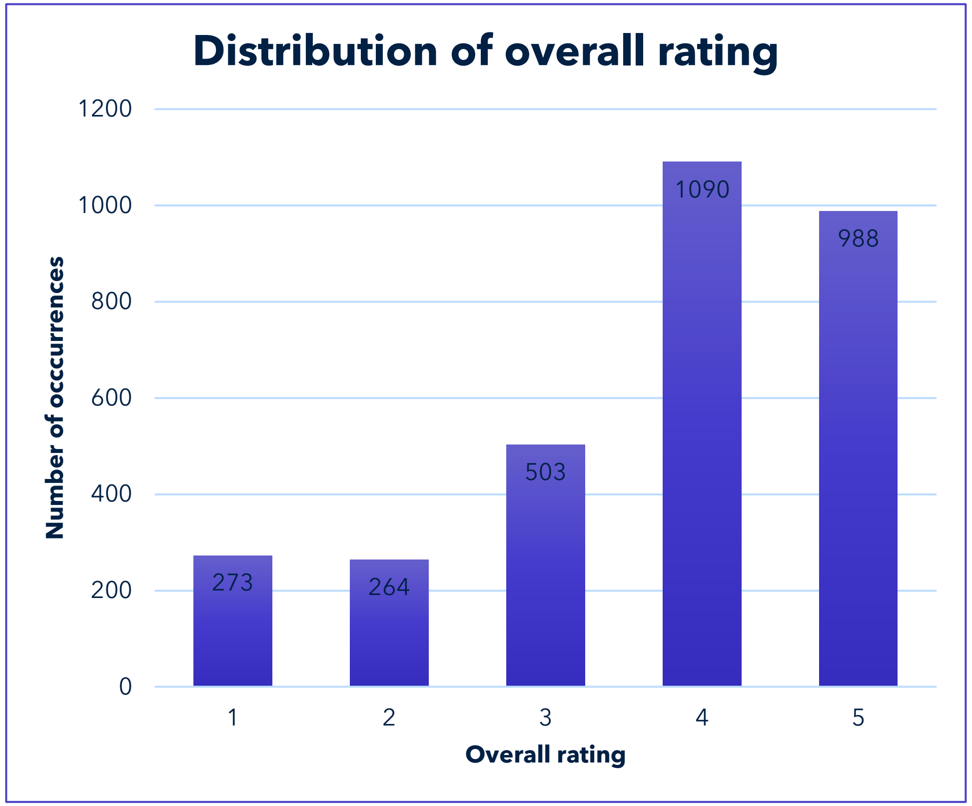
- Before training any machine learning models, the following data preprocessing steps were done:
  - Attributes *Hotel_id*, *id*, *author_id* and *Author_username* were removed as any IDs would not match with the one's in the *Facebook* dataset. In addition, attribute *id* (ID of the review) would not have any predictive power in any case.
  - All missing numeric attribute values were replaced with the mean value of that attribute in the given dataset.
- Case specific data preprocessing steps are presented later in their respective section.

Used Machine learning algorithms

- In all 4 classification cases, the same 5 algorithms were used:
  - **ZeroR** - dummy classifier, which always predicts the majority class. Used as the benchmark for the rest.
  - **J48 decision tree** – non-parametric supervised learning method
  - **Random forest** – combination of multiple decision trees to increase robustness
  - **Naive Bayes** – simple and quick classifier using Bayes' rule
  - **Support Vector Machine** – max-margin model aiming to find rules to separate classes

Dataset

- Dataset contains 3118 reviews
- Two independent variables:
  - text (string)
  - title (string)
- Both text and title contain writing in several languages, but majority language is English
- Dependent variable: *Rating_overall*
- Most ratings either 4 or 5



Case 1: Predicting hotels' overall rating using textual feedback
Analysis description

Target

Predict the overall rating (1 to 5) of the hotel in each review, using ***title*** and ***text*** attributes.

Data preprocessing

- Rating overall converted from numerical to nominal
- Attributes *title* and *text* split jointly to numeric attributes describing the occurrence of each attribute
  - Minimum word occurrence 20
  - LovinsStemmer, MultiStopwords handler, WordTokenizer

Classification configurations

- Cost sensitive classification
- No class balancing
- 10-Fold Cross-Validation was used to minimize model bias and over-fitting
- Symmetric absolute distance-based cost matrix:
  - Predicting too low or high equally costly
  - Cost increases linearly with the error

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	0	1
4	3	2	1	0

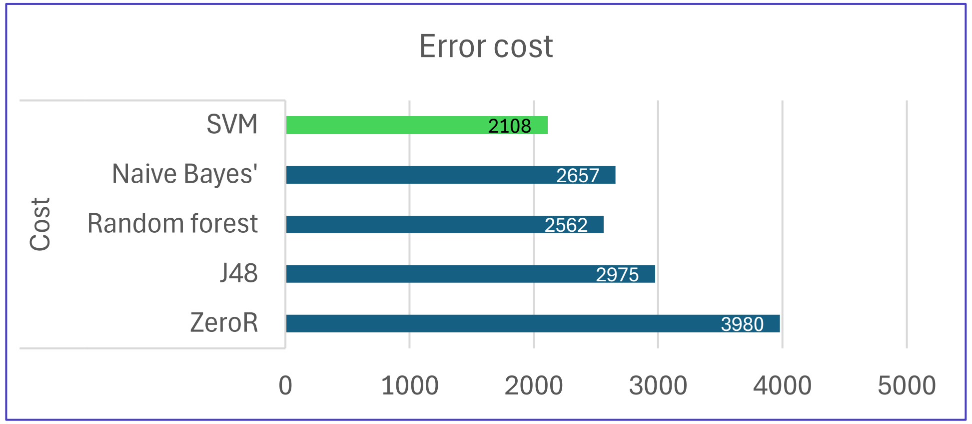
Evaluation criteria

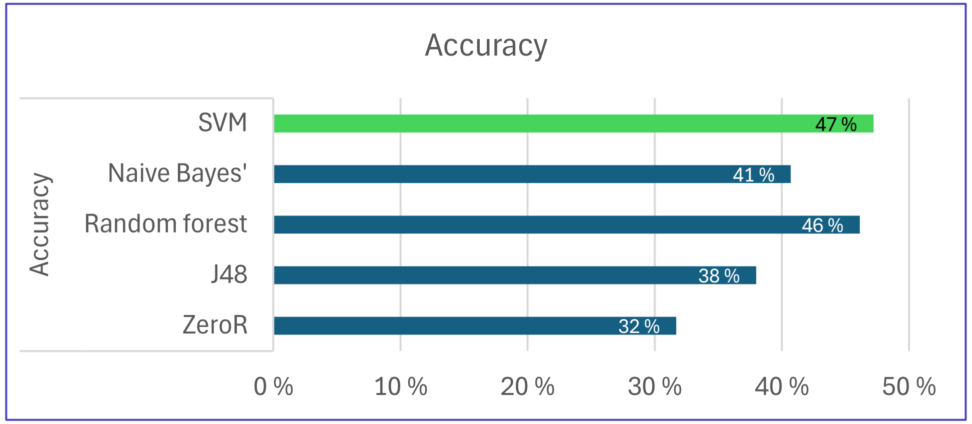
- **Accuracy** (correct predictions over all predictions) to measure the overall hit rate
- **Cost** (according to the cost matrix)

Case 1: Predicting hotels' overall rating using textual feedback
Results

- Each 4 classification methods outperform the benchmark ZeroR.
- Support Vector Machine (SVM) showed both the lowest cost and the highest accuracy.
- The confusion matrix and the low error cost show that SVM rarely make significant (>1) rating errors.

SVM confusion matrix		**Predicted overall rating**				
		1	*2*	*3*	*4*	*5*
Actual overall rating	*1*	152	61	33	24	3
	2	65	69	66	51	13
	3	34	73	160	188	48
	4	20	54	193	487	336
	5	10	16	56	303	603

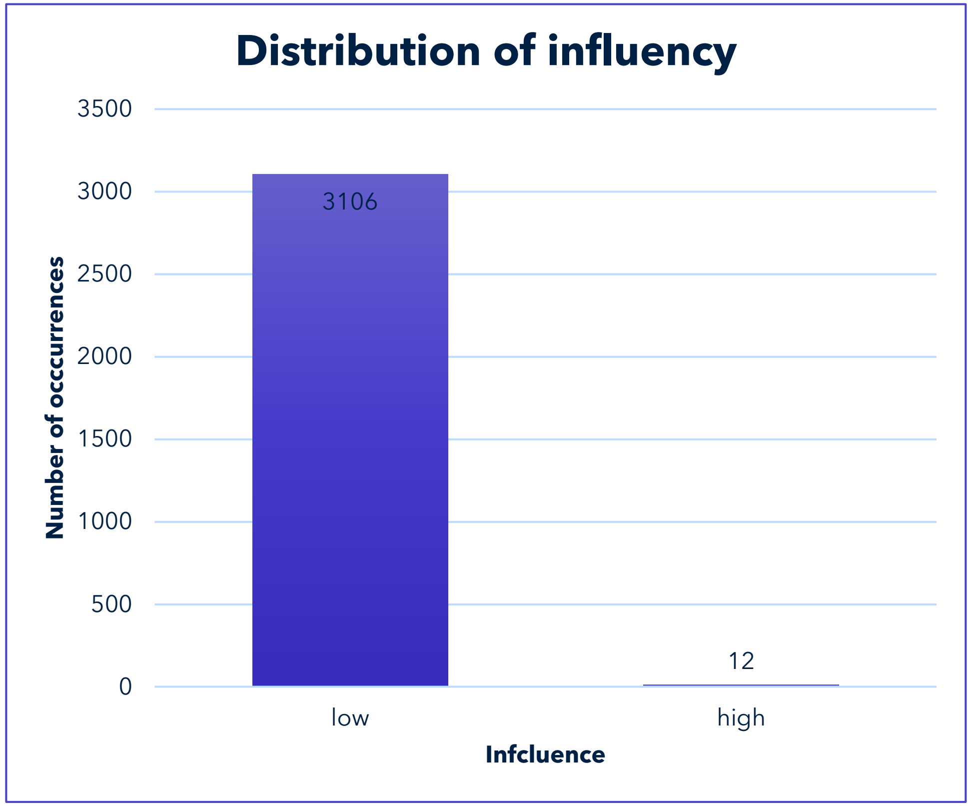




Case 2: Discovering influential reviews based on the textual feedback
Dataset

Dataset

- Dataset contains 3118 reviews
- Two independent variables:
  - text (string)
  - title (string)
- Both text and title contain writing in several languages, but majority language is English
- Dependent variable: *num_helpful_votes* converted into 2-category nominal variable: high (>=15), and low (<15)
- **Only 12 (0,4 %) high-influence reviews!**



Case 2: Discovering influential reviews based on the textual feedback
Analysis description

Target
Predict whether a review is highly influential (over 15 helpful votes), using ***title*** and ***text*** attributes.

Data preprocessing
- Binary nominal variable created based on the values of the *num_helpful_votes* variable
  - High (num_helpful_votes >=15)
  - Low (num_helpful_votes <15)
- Attributes *title* and *text* split jointly to numeric attributes describing the occurrence of each attribute
  - Minimum word occurrence 20
  - LovinsStemmer, MultiStopwords handler, WordTokenizer

Classification configurations
- Class-balancing was used to increase the model sensitivity to high-influence class
- 10-Fold Cross-Validation was used to minimize model bias and over-fitting
- Cost-matrix with very high cost (300) on false-negative was put to put weight on finding the high-influence reviews

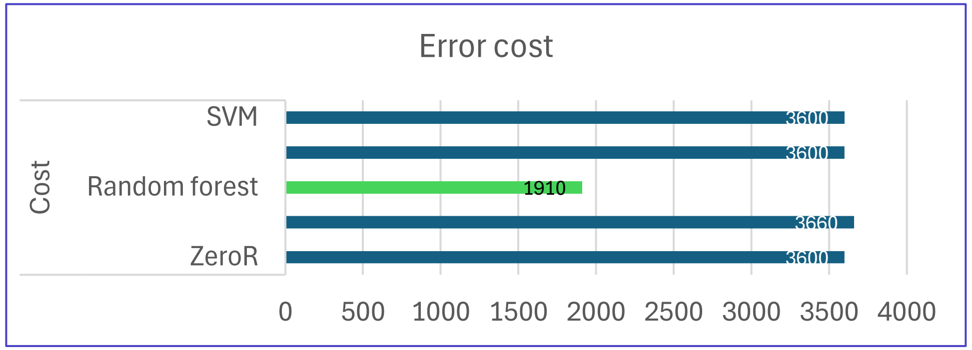
0	1
300	0

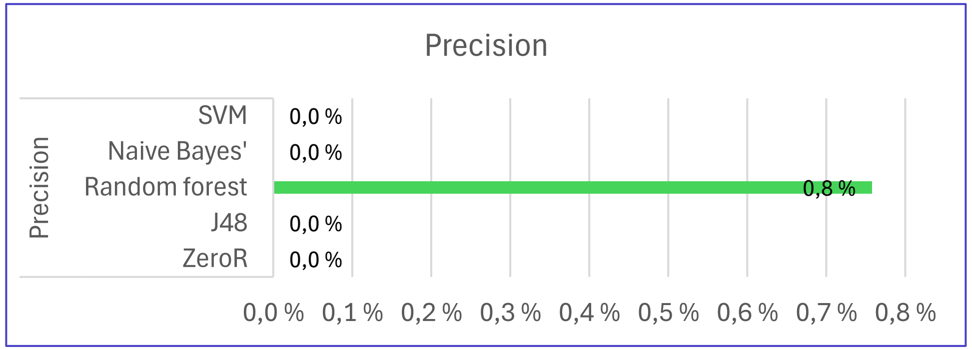
Evaluation criteria
- **Precision** (true positives over all positives) to measure how likely high-influence prediction is correct
- **Recall** (true positives over actual positives) to measure the capture rate of high-influence reviews
- **Cost** (according to the cost matrix)

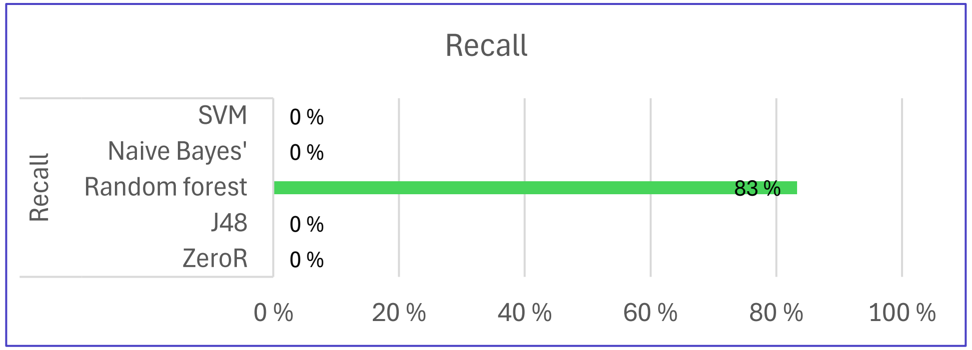
Case 2: Discovering influential reviews based on the textual feedback
Results

- Only Random Forest (RF) was able show any predictive power.
- RF has high capability of finding the actual high-influence reviews (recall 83 %).
- However, it has also several false positives (precision only 0,8%).

RF confusion matrix		Predicted influence	
		low	*high*
Actual influence	*low*	1796	1310
	high	2	10



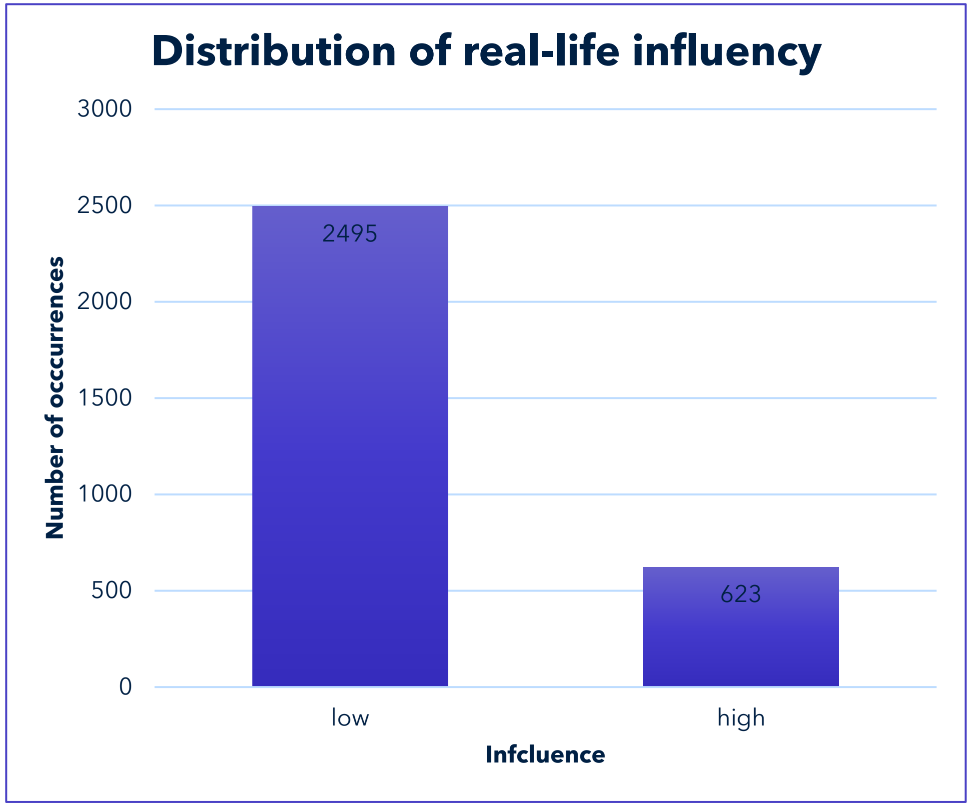




Case 3: Discovering real-life influential authors based on their full feedback
Dataset

Dataset

- Dataset contains 3118 reviews
- 15 independent variables:
  - Afore mentioned 4 IDs and text variables were removed
- Dependent variable: *Author_num_cities* converted into 2-category nominal variable: high (>=15), and low (<15)
- 623 (20 %) real-life high-influence authors.



Case 3: Discovering real-life influential authors based on their full feedback
Analysis description

Target
Predict whether a review author is highly influential in real-life (over 15 cities visited)

Data preprocessing
- Binary nominal variable created based on the values of the *Author_num_cities* variable
  - High (*Author_num_cities* >=15)
  - Low (*Author_num_cities* <15)
- Missing class attribute values replaced with "Low" as it is the majority class.

Classification configurations
- Cost sensitive classification
- No class balancing
- 10-Fold Cross-Validation was used to minimize model bias and over-fitting
- Cost-matrix was used to give false negatives higher cost in comparison to false positives

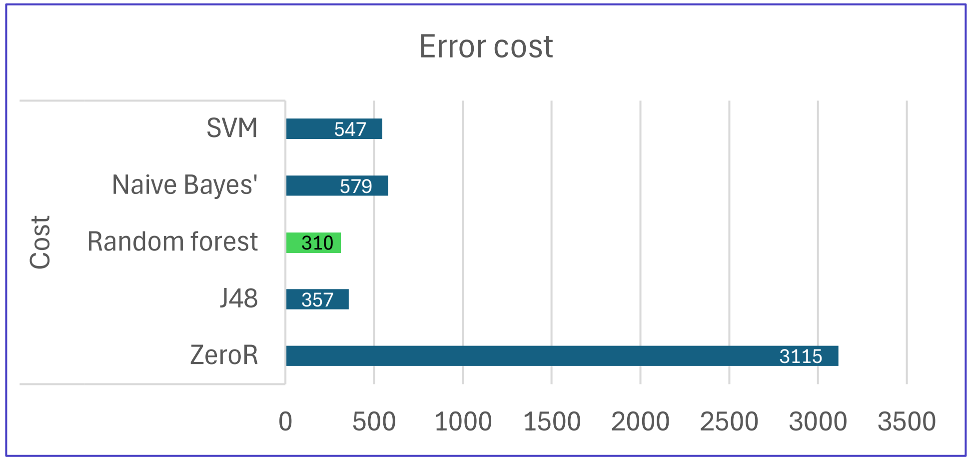
0	1
5	0

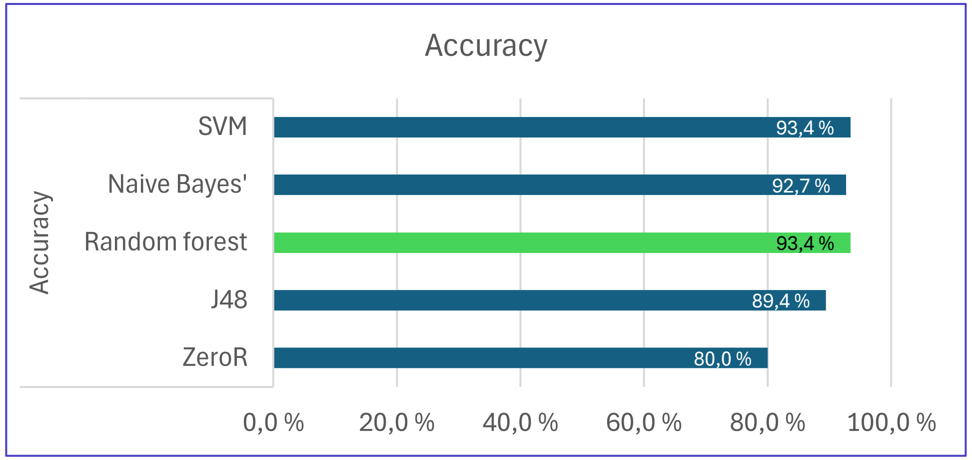
Evaluation criteria
- **Accuracy** (correct predictions over all predictions) to measure the overall hit rate
- **Cost** (according to the cost matrix)

Case 3: Discovering real-life influential authors based on their full feedback
Results

- All 4 models were able to provide high accuracy (~90 % or more)
- Random Forest (RF) showed both the lowest cost and highest accuracy

RF confusion matrix		Predicted influence	
		low	*high*
Actual influence	*low*	2315	180
	high	26	597

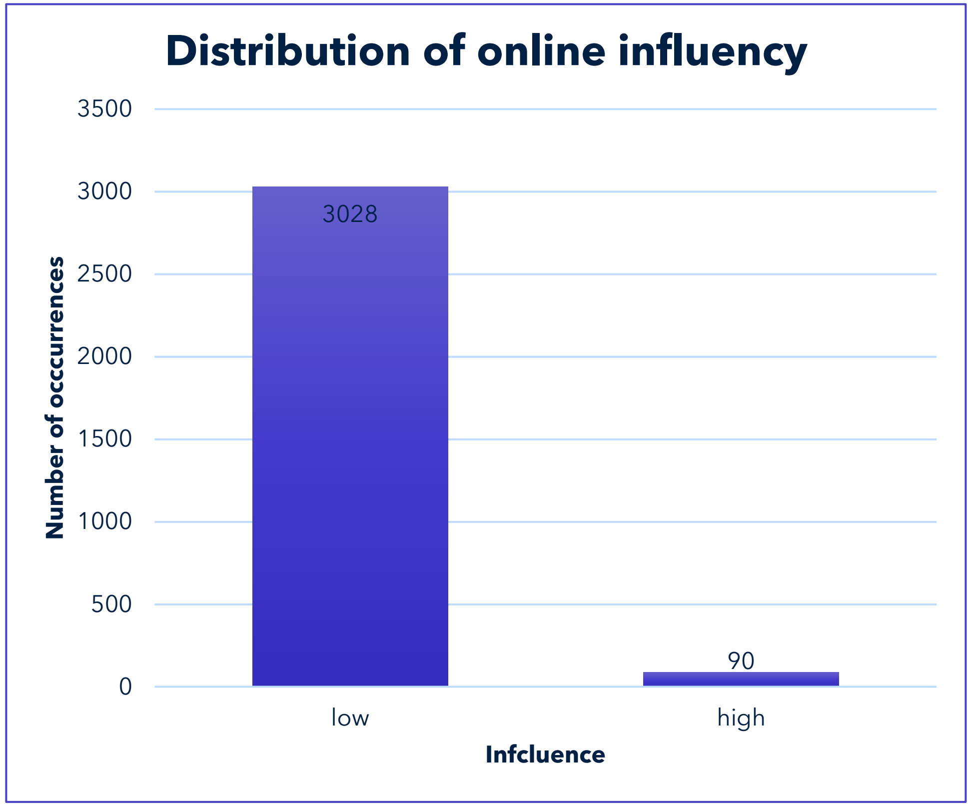




Case 4: Discovering online influential authors based on their full feedback
Dataset

Dataset

- Dataset contains 3118 reviews
- 15 independent variables:
  - Afore mentioned 4 IDs and text variables were removed
- Dependent variable: *Author_num_helpful_votes* converted into 2-category nominal variable: high (>=100), and low (<100)
- **Only 90 (2,9 %) online high-influence authors!**



Case 4: Discovering online influential authors based on their full feedback
Analysis description

Target

Predict whether a review author is highly influential in online community (over 100 review helpfulness votes)

Data preprocessing

- Binary nominal variable created based on the values of the *Author_num_helpful_votes* variable
  - High (*Author_num_helpful_votes* >=100)
  - Low (*Author_num_helpful_votes* <100)
- Missing class attribute values replaced with "Low" as it is the majority class.

Classification configurations

- Cost sensitive classification
- No class balancing
- 10-Fold Cross-Validation was used to minimize model bias and over-fitting
- Cost-matrix was used to give false negatives higher cost in comparison to false positives

0	1
5	0

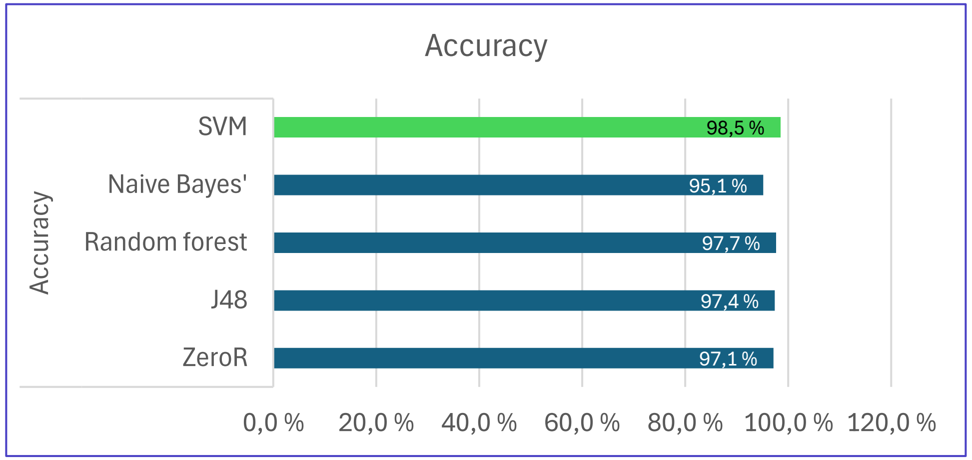
Evaluation criteria

- **Accuracy** (correct predictions over all predictions) to measure the overall hit rate
- **Cost** (according to the cost matrix)

Case 4: Discovering online influential authors based on their full feedback
Results

- SVM, RF and J48 were able to provide higher accuracy than the benchmark.
- Random Forest (RF) showed the lowest cost
- SVM showed the highest accuracy
- Interestingly, Naibe Bayes' showed lowest recall (not visible in graph) yet lowest accuracy

RF confusion matrix		**Predicted influence**	
		low	*high*
Actual influence	*low*	2985	43
	high	30	60



Recommendations

Short-term

1. Start using the Support Vector Machine model for the case 1.
2. Do not use any of the tested models for the case 2
3. Start using the Random Forest model for the cases 3.
4. Use either SVM or RF model to the case 4

Mid-term

- **Cases 1, 3 and 4:** Continue iterative ML development with the most promising algorithms: RF and SVM
    - Put significantly more focus on the attribute selection
- **Case 2:** continue investigations to find a reasonable prediction accuracy
    - Try different ways to split text into numeric variables
    - Put emphasis to attribute selection to reduce the amount of word attributes.