Using Weka 3 to Enhance Competitiveness at Horizon Hotels

Organization Overview

This analysis focuses on a fictional hotel chain, *Horizon Hotels*, which operates both city and resort properties across various locations. The company serves a diverse customer base, including business and leisure travelers, and competes by offering a blend of comfort, convenience, and value. Horizon Hotels utilizes multiple distribution channels, including travel agents, corporate partnerships, and direct online bookings. With growing competition in the hospitality industry, the company seeks to improve its competitiveness by leveraging data to better understand its customers, tailor its services, and increase profitability.

Strategic Questions to Enhance Competitiveness

To remain competitive and improve decision-making, Horizon Hotels must address several key business questions:

- Who are our main customer segments, and how do their behaviors differ?
- Can we predict which reservations are likely to be canceled?
- What services or amenities tend to be booked together?
- How do users typically navigate our website before completing a booking?
- What are guests saying about their experiences, and how can we use that feedback to improve satisfaction?

These questions, when answered effectively, can lead to more targeted marketing, improved guest experiences, optimized pricing strategies, and increased customer retention.

How Weka 3 Supports Data Mining Techniques

Weka 3 offers data mining tools that can assist Horizon Hotels in addressing its strategic questions. Applying Weka involves several key steps: first, **data preprocessing**, which includes cleaning the data and formatting it into CSV or ARFF files; second, **applying a selected model** to the dataset; third, **testing various algorithms** to identify the most effective one; and finally, **evaluating model performance** using metrics such as accuracy, precision, recall, and confusion matrices to determine the most suitable approach for the business objective.

Weka supports various techniques including clustering, prediction, association rule mining, and sequential analysis.

Clustering

Clustering is applied when there are no predefined outcomes or labels in the data. It is useful for discovering natural groupings or hidden patterns, such as identifying different types of customers based on their booking behavior, travel season, or spending habits.

Example question: *What are the main types of customers we serve?*

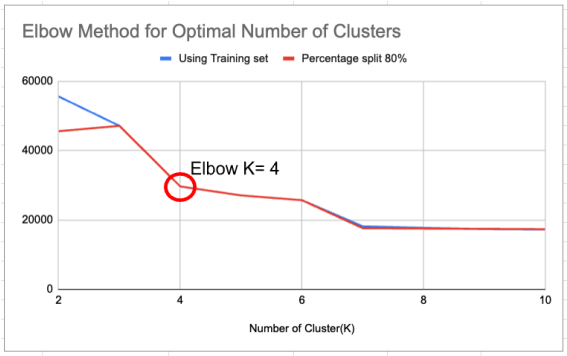
To explore different segmentation outcomes and compare results, multiple clustering models can be tested in Weka. In this analysis, I used two clustering algorithms: **SimpleKMeans** and **Cobweb (CM)**, both available in Weka, to uncover distinct customer segments and evaluate their characteristics.

Data Preprocessing: Standardization

All numeric attributes were standardized to ensure fair comparison during clustering. However, to avoid distorting binary attributes such as is_canceled and is_repeated_guest, these fields were first converted from numeric to nominal. After standardizing the remaining attributes, the binary fields were converted back to numeric format to retain their original meaning and usability for further analysis.

Clustering with K-Means

SimpleKMeans clusters data based on Euclidean distance, assigning each data point to the nearest cluster centroid. I experimented with values of **K from 2 to 10** to find the optimal number of clusters. To evaluate this, I used the **Elbow Method**, which analyzes the **Within-Cluster Sum of Squares (WCSS)** — a metric automatically calculated by WEKA.
I exported WCSS values to **Google Sheets** and plotted them with K on the X-axis and WCSS on the Y-axis. The elbow point, where the WCSS sharply decreases before leveling off, was observed at **K = 4**. This indicated that **four clusters** provide the best balance between model complexity and explanatory power.



To ensure robust results, I tested the clustering on both the **full dataset** and an **80% training set split**, a common machine learning practice. Both setups produced similar outcomes, indicating that the data is **stable and consistent** across different training conditions.

K-Means Clustering Results and Customer Segments

The K-Means model identified **four distinct customer segments**, each reflecting unique booking patterns, stay behavior, and revenue contribution:
Cluster 1 (12%):
City Hotel, very low cancellation rate (~8.6%), lead time is very short (~51 days), channel is **Direct**, repeat guests is high (0.129), ADR is high (104.32), stay period is relatively short (~3 days), and revenue per booking is **346**.
Likely frequent, loyal business travelers who book directly and don't cancel. These are **high-value retention targets** — ideal for **loyalty rewards** and **convenience-focused services**.
Cluster 2 (28%):
Resort Hotel, cancellation rate is moderate (~33%), lead time is mid-range (~99 days), channel is mostly **Agent**, repeat guests are low (0.0366), ADR is moderate (93.51), stay period is the longest (~4.5 days), and revenue per booking is the **highest (441)**.
Likely leisure travelers or vacationers who book through agents. While they stay longer and spend more, they have **higher cancellation risk**. Ideal for **upsell offers**, **flexible booking terms**, and **reminders** to reduce cancellations.
Cluster 3 (27%):
City Hotel, cancellation rate is **100%**, lead time is very long (~154 days), channel is **Agent**, repeat guests are very low (0.0124), ADR is high (104.38), stay period is short (~3 days), and revenue per booking is **326**.
These are **high-risk or potentially non-genuine bookings** — likely made far in advance and always canceled. This group requires **investigation** and potentially **stricter policies** such as **deposits**, **confirmation steps**, or **channel restrictions**.
Cluster 0 (33%):

City Hotel, no cancellations (0%), lead time is moderate (~87 days), channel is **Agent**, repeat guests are very low (0.0319), ADR is high (105.91), stay period is short (~3 days), and revenue per booking is **317**. **Reliable one-time guests**, likely **business or package travelers** who book through agents. Although they don't cancel, they aren't loyal customers. This group could be targeted with **retention marketing** and **value-added offers** to convert them into return guests.

```
Attribute                 Full Data           0           1           2           3
                         (119388.0)    (39757.0)   (14240.0)   (33521.0)   (31870.0)
===============================================================================================
hotel                    City Hotel   City Hotel   City Hotel Resort Hotel   City Hotel
is_canceled=1                0.3704            0      0.0865      0.3318           1
lead_time                  104.0054      87.2834      50.757     99.3512      153.553
distribution_channel          Agent        Agent      Direct       Agent        Agent
is_repeated_guest=1          0.0319       0.0089       0.129      0.0366       0.0124
adr                         101.8329      105.914    104.3241     93.5118     104.3809
stay_period                   3.4279       2.9935      3.0027      4.4914       3.0411
revenue_per_booking         357.8547     317.0683    346.3163    441.4569     325.9573



Time taken to build model (full training data) : 0.2 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       39757 ( 33%)
1       14240 ( 12%)
2       33521 ( 28%)
3       31870 ( 27%)
```

Clustering with EM (Expectation-Maximization)

I also applied the **EM algorithm** under two conditions to see how it compared to K-Means:

Condition 1: Let WEKA Determine K Automatically

I set **numClusters = -1** to allow WEKA to determine the optimal number of clusters using cross-validation. The result was a segmentation into three customer segments: Cluster 0 – 79%, Cluster 1 – 20%, Cluster 2 – 2%

This result suggests that most customers fall into **one dominant group**, while a small minority form **niche segments**. The 2% cluster likely represents a small, specialized group or outliers.

Condition 2: Manually Set K = 4

When I manually set the number of clusters to 4, the distribution became very uneven: Cluster 0 – 8%, Cluster 1 – 65%, Cluster 2 – 27%, Cluster 3 – 0%

One cluster contained n**o meaningful data**, indicating that the model may have **overfit or failed to find four natural groups** in the data.

The EM clustering results indicate that the **data is unbalanced**, with one or two **very small clusters**. This suggests the presence of niche behaviors or noise, and highlights the **sensitivity of EM** to uneven data distribution. While EM offers probabilistic insights, the **K-Means model produced clearer, more actionable segments** in this case.

By selecting the most effective clustering model—such as K-Means with four distinct customer segments—Horizon Hotels can tailor marketing, pricing, and operational strategies to each group's behavior. High-value, loyal guests can be rewarded to encourage repeat bookings, while high-risk segments prone to cancellations can be targeted with flexible policies or engagement offers. These insights, combined with future use of Weka's predictive and association tools, can help Horizon Hotels make smarter, data-driven decisions that improve customer satisfaction, reduce risk, and enhance overall competitiveness.

Prediction

Prediction techniques (like decision trees or regression) use historical data to forecast future outcomes, such as whether a booking will result in a no-show or cancellation.
Example question: Can we predict which bookings are likely to be canceled?

To predict which bookings are likely to be canceled, we can make a classification using supervised learning in WEKA. For the first run, I used the simplest model, **J48**, which is a decision tree algorithm. I applied it using three different evaluation methods: full training set, 80% training/test split, and 10-fold cross-validation. The results are as follows: **Accuracy**: 80.95%, 76.58%, 77.45%, **ROC**: 0.86, 0.81, 0.82, **F1 Score**: 0.69, 0.61, 0.63. These results indicate that the J48 model performs reasonably well. However, the high accuracy from the full training set suggests overfitting—it memorizes patterns from the training data rather than generalizing to new, unseen data. Therefore, the full training set result should be viewed as a reference only, not a reliable performance measure.

To compare, I also tested **logistic regression** and **random forest** models. The logistic regression model demonstrated lower performance: **Accuracy**: ~67.3%, **ROC**: ~0.7, **F1 Score**: ~0.42
This suggests that logistic regression may struggle to capture complex patterns in the data compared to tree-based methods.
The **random forest model**, tested with 10-fold cross-validation, produced strong and balanced results: **Accuracy**: 79.7%, **ROC**: 0.86, **F1 Score**: 0.711

Model	Accuracy (%)	ROC	F1 Score
J48 - full training set	80.95	0.862	0.69
J48 - split 80% train	76.58	0.81	0.61
J48 - 10 fold cross validation	77.45	0.82	0.63
Logistic - split 80%	67.27	0.69	0.42
Logistic - 10 fold cross validation	67.42	0.70	0.425
Random Forest - 10 fold cross validation	79.70	0.86	0.711
NaiveBayes - 10 fold cross validation	66.90	0.68	0.422

```
RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 24.12 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       95163               79.709  %
Incorrectly Classified Instances     24225               20.291  %
Kappa statistic                          0.5551
Mean absolute error                      0.2455
Root mean squared error                  0.376
Relative absolute error                 52.6293 %
Root relative squared error             77.863  %
Total Number of Instances           119388

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.871    0.328    0.819      0.871    0.844      0.557    0.861     0.895     0
               0.672    0.129    0.753      0.672    0.711      0.557    0.861     0.832     1
Weighted Avg.  0.797    0.254    0.794      0.797    0.794      0.557    0.861     0.872

=== Confusion Matrix ===

     a     b   <-- classified as
 65435  9729 |     a = 0
 14496 29728 |     b = 1
```

This shows that random forest outperforms the other models in both accuracy and robustness. It also generalizes better than J48 trained on the full dataset, making it the most reliable model among those tested. Still, running a random forest with an **80% training/test split** could be useful to further validate its consistency.
Once the best model is selected, we can use it to **predict which bookings are likely to be canceled**. To do this, new data should be preprocessed in the same format used for training the model. The predictions can then guide proactive actions, such as sending reminder emails, offering flexible cancellation policies, or providing targeted promotions to reduce cancellation likelihood.
Collectively, these data-driven strategies can empower **Horizon Hotels** to make smarter operational decisions, minimize cancellations, and maintain a competitive edge in the hospitality industry.

Association

Association rule mining is used to identify frequent co-occurrences between variables. This technique is valuable for discovering which services or amenities are often booked together by guests.
Example question: *What services or products are frequently booked together by our guests?*
In Weka, we can use the **Apriori** algorithm to uncover frequent itemsets and generate association rules. The process follows the same steps as before: prepare the data, load it into Weka, run the association rule mining, and interpret the results to identify meaningful patterns and the best model.

Sequential Analysis

Sequential analysis looks at the order in which events occur, which is particularly valuable for analyzing user behavior on the hotel's website.
Example question: *What are the most common browsing paths users follow before making a reservation?*
While Weka doesn't natively support advanced sequential pattern mining, a similar effect can be achieved by preprocessing web session data to reflect sequences of actions, such as page views or clicks. Once structured properly, specialized plugins or external tools can be used in conjunction with Weka to extract common sequences. This helps the hotel better understand customer behavior and optimize website design, content placement, or marketing strategies to encourage conversions.

Data Requirements and Preparation

To successfully use Weka for these analyses, datasets need to be clean and formatted in **CSV** or **ARFF** files. The data can include **structured data**, such as spreadsheet data, transaction records, and booking logs; **unstructured data**, such as customer reviews, chat logs, email inquiries, and survey responses; and **web usage data**, such as clickstream paths, page view sequences, and session durations.
In the example above, I used structured data and applied data mining techniques to extract valuable insights aimed at increasing the hotel's competitiveness. However, if I were working with unstructured data like customer reviews, I could apply **text mining techniques** using Weka to identify recurring themes, sentiments, and guest concerns. These insights could help the hotel understand customer experiences more deeply and prioritize service improvements. Similarly, if I had access to web usage data, **web mining techniques** could be used to analyze user behavior on the website, helping to optimize the booking process and enhance the overall user experience.

Weka is a beginner-friendly data mining tool that supports a wide range of tasks, including data mining, unstructured data processing, and even basic image analysis. To improve model performance, it is essential to understand how the model works, tune its parameters effectively, and interpret evaluation metrics accurately.

Weka's user-friendly interface makes it accessible for those new to data mining. Additionally, it integrates well with other tools such as R and allows for further customization of models through Java programming.