Unlocking insights

Hotel Rating Classification for Enhanced Customer Experience

Group 22

BUSINESS PROBLEM & MOTIVATION AND AIMS OF THE STUDY

BUSINESS PROBLEM & MOTIVATION

Hotels highly value customer reviews as they help understand strengths and weaknesses in their services. Positive reviews **attract more customers** and boost profits while negative reviews **suggest opportunities for improvement**.

- Manual analysis of each review is **labour-intensive** and **inefficient**.
- It is difficult to capture a **whole picture** of customer reviews by means of manual analysis.

AIMS OF THE STUDY

- The goal of the project is to build a robust model that **classifies customer ratings** as **High** (4-5) or **Low** (0-3) using machine learning, reducing the need for manual review.
- Our focus is to correctly predict **low ratings** (True Negative cases) because it allows for **early identification** of service failures and **timely intervention** in saving the public image and improving guest experience.
- This model can be used by different stakeholders for different purposes.
  - **Hotels** themselves benefit most from the model because they can have better insights into not only **how (dis)satisfied their customers are** but also **how well their rivals have performed**.
  - **Guests** may also find the model useful as it helps understand **what they can reasonably expect** from staying in a (specific) hotel.

DATASET OVERVIEW

DATA SOURCE

- The dataset is sourced from a large open TripAdvisor review dataset available on **Kaggle**. The dataset consists of 878,561 rows and 19 attributes. To reduce processing time and computational cost, we have chosen to work with a **subset** of these instances.
- The dataset includes two files 'offerings.csv' and 'reviews.cvs' which are merged using a shared column 'offering_ID' which represents the hotels' ID.
- Among the available variables, the **target variable** is 'overall_rating'.
- Web scraping was used to gather additional information about the hotels, including details such as the hotel class, number of amenities, price range, distance to the nearest metro station, and the number of nearby attractions, among other features.

→ **The final dataset (after data collection and merging) has 239,166 rows and 16 attributes.**

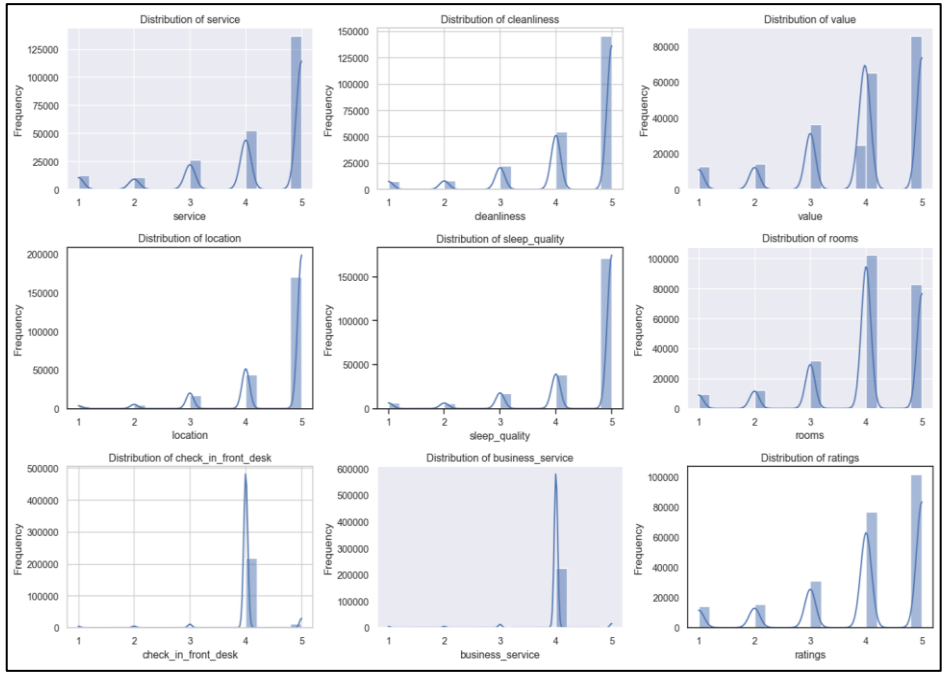
DATA PREPROCESSING

Sentiment labelling: Label each review based on its star rating. One to three (1-3) is treated as **negative** (**low**) while four to five as **positive** (**high**). The reasons for a **binary classification** include:

- **Simplified analysis**: Classification as 'high' and 'low' allows for a more intuitive interpretation of the results, i.e. whether the customer is either satisfied or dissatisfied.
- **Actionable insights**: Binary grouping enables targeted actions based on whether are experiencing positive or negative experiences, rather than focusing on a specific rating value.
- **Reduced noise**: Treating the target as a numerical variable can introduce noise as there may not be a clear distinction between adjacent rating values (e.g. 3 vs. 4).
- **Improved performance**: Binary classification models are often simpler to train, require fewer computational resources, and can achieve higher performance.

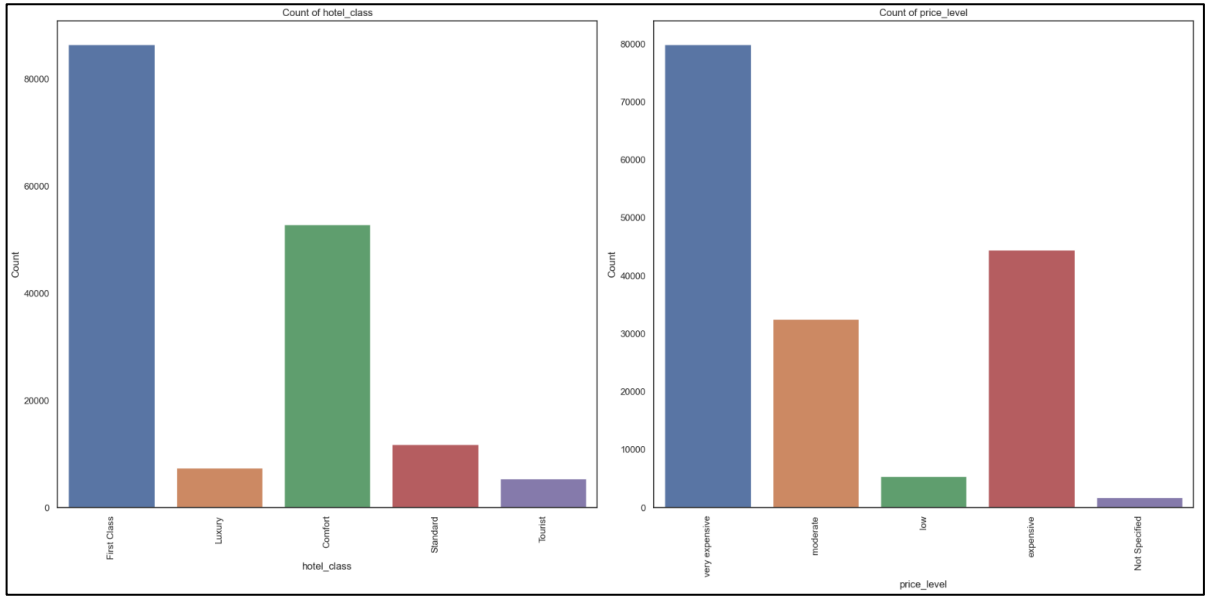
EXPLORATORY DATA ANALYSIS (EDA)

NUMERICAL



- Ratings range from 1 to 5.
- Most 5s are found in **service**, **cleanliness**, **value**, **location**, and **sleep quality** while **rooms**, **front desk service**, and **business services** are mostly rated 4.
- **Highest ratings are primarily 5s**, followed by 4s.
- Distributions are **left-skewed**, with most ratings on the higher end and a long tail toward lower values.

CATEGORICAL



- **Most hotels are categorised as 'First Class,'** followed by 'Comfort' as the next most common category. The other hotel classes fall under **relatively the same range** of number.
- Correspondingly, the **'Very Expensive' price level comprises the largest proportion**, followed by 'Expensive' and 'Moderate'. The remaining price levels witness **insignificant numbers** compared to higher price levels.

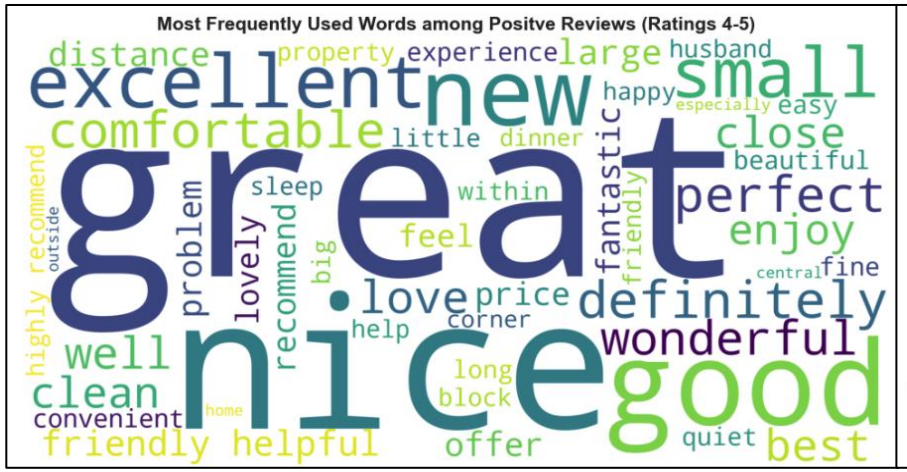
EXPLORATORY DATA ANALYSIS (EDA)

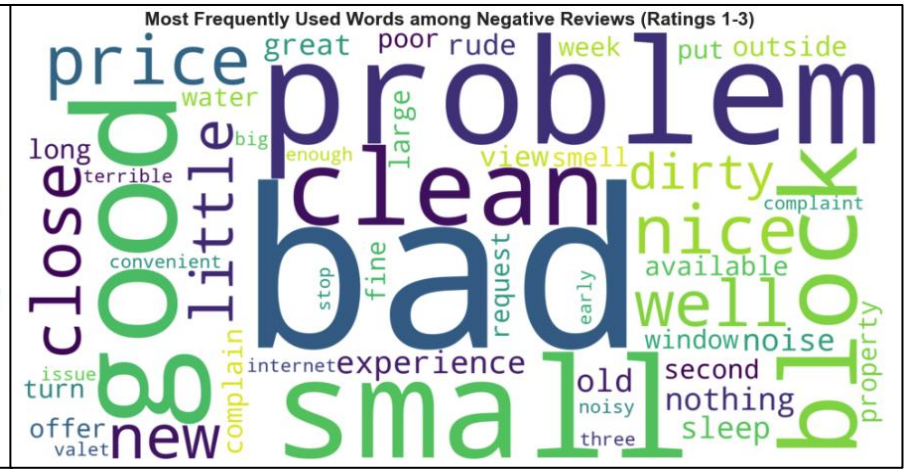
NUMERICAL

The two figures show the number of meaningful words found in negative reviews and positive reviews.

- Both distributions are **right-skewed**.
- Few posts have over 1,000 characters or 200 meaningful words.
- Only a small percentage of positive reviews exceed 200 meaningful words.
- The average number of meaningful words in negative reviews is 56, while positive reviews have 70.
- Positive reviews show higher variance in the number of meaningful words, indicating that satisfied customers tend to write longer reviews.

CATEGORICAL





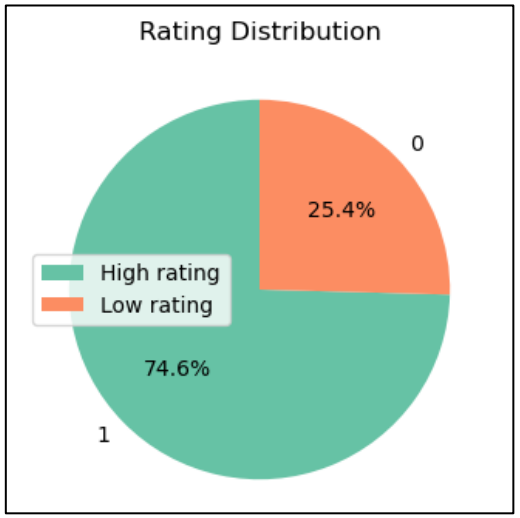
The **larger the word**, **the higher its frequency** of appearance in the reviews. Terms like ‘bad,’ ‘problem,’ and ‘poor’ represent the negative sentiment class. Words such as ‘great,’ ‘nice,’ ‘excellent,’ ‘good,’ ‘clean,’ and ‘new’ represent the positive sentiment class.

DATA PROCESSING PROCEDURES

1. TARGET VARIABLE BINARY TRANSFORMATION The original customer review ratings (0-5 scale) have been grouped into two categories: Low (below 4) and High (4-5), enabling binary classification.

Why binary classification? (RECALL)

- **Simplified Analysis:** Easier interpretation as "low" vs. "high" ratings
- **Actionable Insights:** Help to identify customers at risk of churn (low ratings)
- **Reduced Noise:** Grouping minimises variability between adjacent ratings.
- **Improved Performance:** Simplifies model training and increases accuracy.



There is **an imbalance in the dataset:** 25.4% of ratings are classified as Low, while 74.6% are classified as High.

The goal is to **predict the minority class (low ratings),** which is essential for identifying potential customer dissatisfaction.

2. NOMINAL CATEGORICAL VARIABLES

Apply one-hot coding to 'price level'.

price_level_Not Specified	price_level_expensive	price_level_low	price_level_moderate	price_level_very expensive
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1
0	0	0	0	1

3. NUMERICAL VARIABLES Use StandardScaler to standardise features with high variance and high skewness (above 0.5).

```
print(variance)
```

hotel_class	9.154227e-01
no_of_amenities	5.170557e+02
dist_to_metro	4.473735e+04
no_of_attractions	2.559658e+00
no_of_reviews	1.251214e+07
service	1.293059e+00
cleanliness	1.030005e+00
value	1.226482e+00
location	6.614198e-01
sleep_quality	8.633781e-01
rooms	1.055495e+00
check_in_front_desk	1.804572e-01
business_service	1.329029e-01

Skewness of Continuous Numeric Features:

no_of_reviews	1.567536
no_of_amenities	0.640266
no_of_attractions	0.589653
dist_to_metro	0.514933
value	-1.037590
rooms	-1.165960
service	-1.468825
hotel_class	-1.617172
cleanliness	-1.704974
location	-2.150575
sleep_quality	-2.156172
check_in_front_desk	-3.154459
business_service	-4.543579

MODELLING

CHOICE OF MODELS

The goal is to choose the best out of **7 models below** using the original imbalanced data and rebalanced data using resampling methods:

- SMOTE
- ADASYN
- Random Under Sampling methods.

Model	Practical Reasons for Use
Logistic Regression	- Simple, interpretable benchmark for binary classification. - Efficient for large datasets; quick insights on predicting low ratings.
Decision Tree	- Captures non-linear patterns influencing low/high ratings. - Provides visualization for decision paths, aiding interpretation.
Random Forest	- Reduces overfitting by averaging multiple trees. - Identifies key features driving low ratings, aiding targeted actions.
Naive Bayes	- Efficient with large datasets; handles categorical features well (e.g., price level). - Suitable for predicting minority class in skewed data.
Ensemble Methods	- **Bagging (Random Forest)**: Reduces variance, improving stability. - **AdaBoost**: Focuses on misclassified low ratings to enhance prediction. - **XGBoost**: Optimized for large datasets; effectively handles class imbalance.

MODEL DEVELOPMENT WORKFLOW & DATASET USAGE

The data was split into **60% training**, **20% validation**, and **20% test** sets.

Step	Description	Dataset Used
1. Split Data	Split into **training** (`X_train, y_train`), **validation** (`X_val, y_val`), and **test** (`X_test, y_test`) sets.	Full dataset
2. Resampling Methods	**Address class imbalance** on the **training set** only.	Training set (`X_train, y_train`)
- SMOTE	**Generate synthetic samples** for minority class (low ratings).	Resampled training set
- ADASYN	Similar to SMOTE, with more focus on **underrepresented areas**.	Resampled training set
- Random UnderSampling	**Reduce samples** from the majority class (high ratings).	Resampled training set
3. Train Initial Model	Train using **resampled training set**.	Resampled training set
4. Evaluate Initial Model	Assess on **validation set** to establish a baseline metric (e.g., **accuracy, F1 score**).	Validation set (`X_val, y_val`)
5. Feature Selection	**Select relevant features** using resampled training data.	Resampled training set
6. Train Model with Selected Features	Retrain using **selected features**.	Resampled training set (selected)
7. Evaluate Model	Compare to baseline on the **validation set**.	Validation set (`X_val, y_val`)
8. Hyperparameter Tuning	Optimize model parameters, validate on **validation set**.	Training & Validation sets
9. Test Final Model	**Evaluate on test set** to measure performance on unseen data.	Test set (`X_test, y_test`)

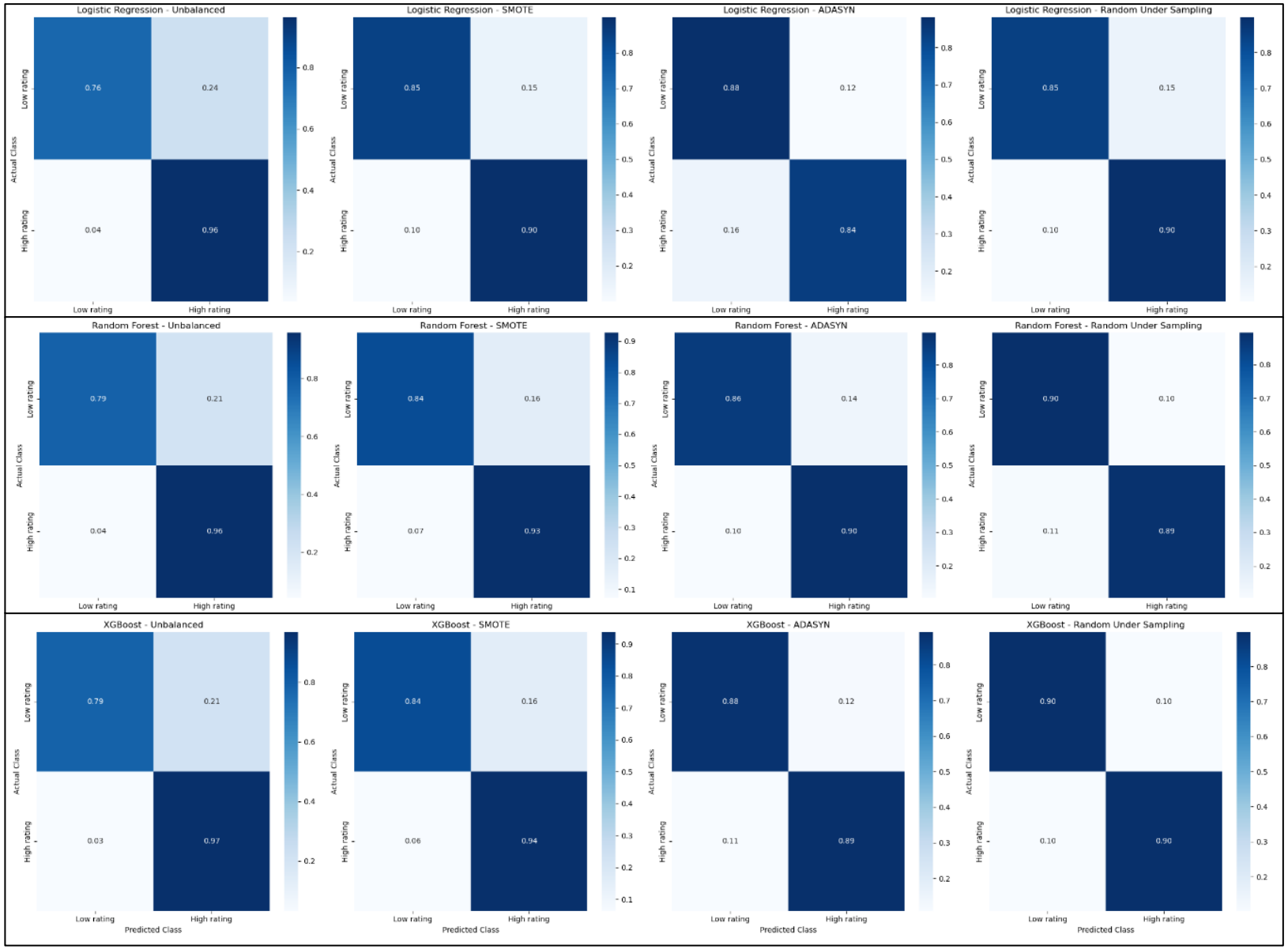
MODELLING

INITIAL MODEL TRAINING

Confusion matrix (all features)

Key performance metric: True Negative rate / Recall rate for 'low rating' class.

- This test was **based on the validation set**, using **all available features** (21 features).
- As can be seen from the confusion matrices, every model's **ability to predict True Negative ('low rating' class) is improved significantly with balanced dataset.**
- **Best Resampling Method**: Among the resampling methods, **Random Under Sampling** consistently improved the recall for predicting "low ratings" across different models. Hence, we will proceed with only the Random Under Sampling method.
- Two best models with highest recall rate for 'low rating' class (True Negative) is **Random Forest** (0.895) and **XGBboost** (0.897).

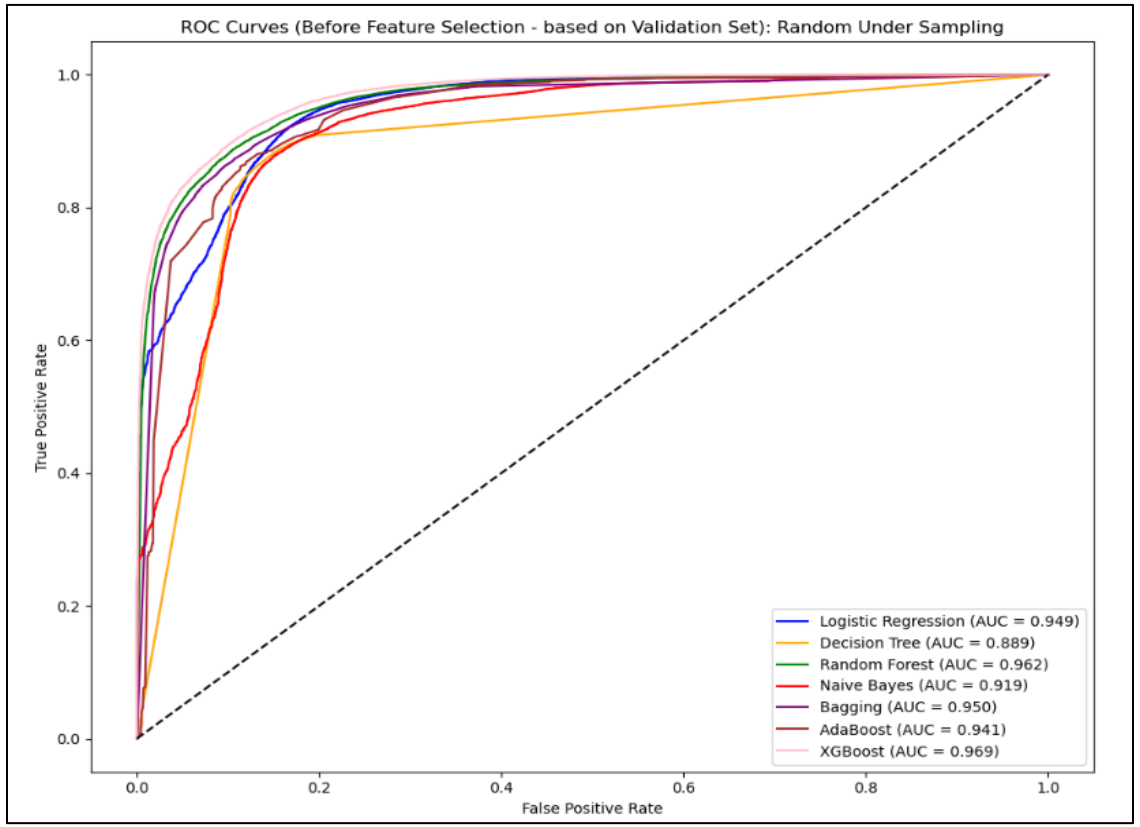


MODELLING

INITIAL MODEL TRAINING

ROC curves & AUC (all features)

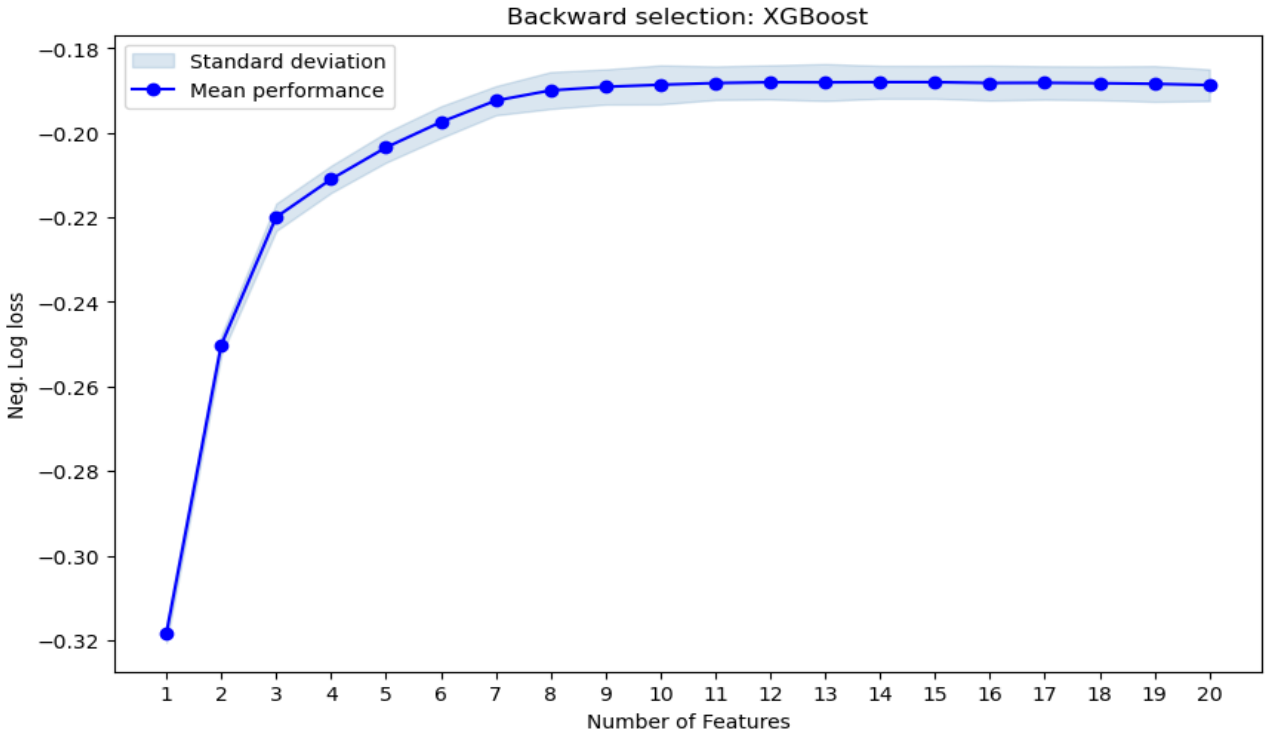
- All models demonstrate strong performance, with **XGBoost achieving the highest AUC (0.970).**
- AUC scores remain stable when transitioning from the unbalanced to the rebalanced dataset, indicating that the models maintain their discriminative ability after resampling.
- While AUC is a crucial metric for assessing model performance, it is **essential to consider other metrics for the minority class**, especially in imbalanced scenarios.
- **XGBoost** consistently shows good performance across all metrics, achieving the highest **accuracy (0.897), recall for the low ratings class (0.897), and a strong F1-score (0.816),** indicating excellent effectiveness in predicting the minority class.



Model	Accuracy	Precision ("low ratings" class)	Recall ("low ratings" class)	F1-score ("low ratings" class)
Logistic Regression - Random Under Sampling	0.886146	0.740232	0.850658	0.791612
Decision Tree - Random Under Sampling	0.864529	0.684224	0.867434	0.765013
Random Forest - Random Under Sampling	0.888738	0.728879	0.895395	0.803602
Naive Bayes - Random Under Sampling	0.882947	0.749867	0.809622	0.778599
Bagging - Random Under Sampling	0.878097	0.705527	0.893339	0.788402
AdaBoost - Random Under Sampling	0.877219	0.712816	0.865872	0.781924
XGBoost - Random Under Sampling	0.897309	0.748730	0.897122	0.816236

MODELLING

FEATURE SELECTION

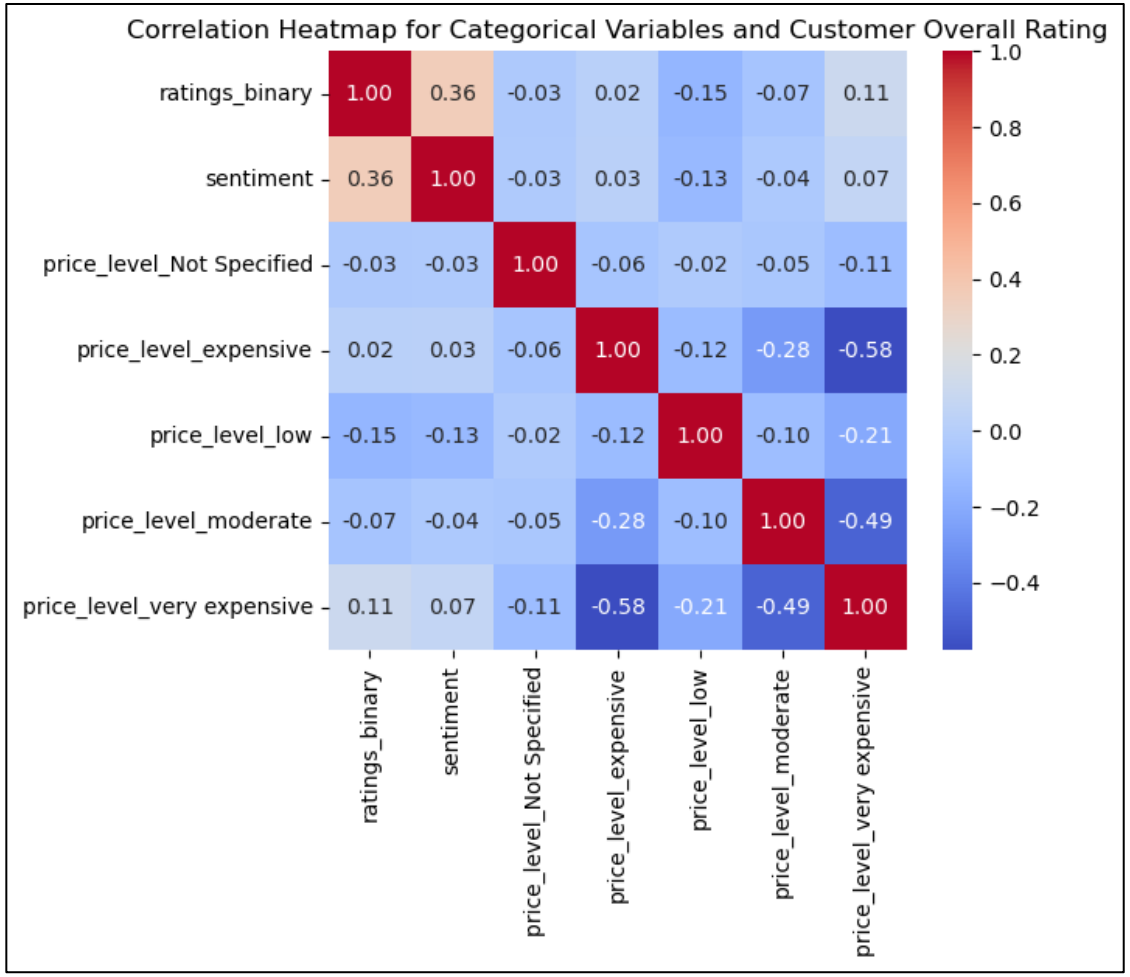


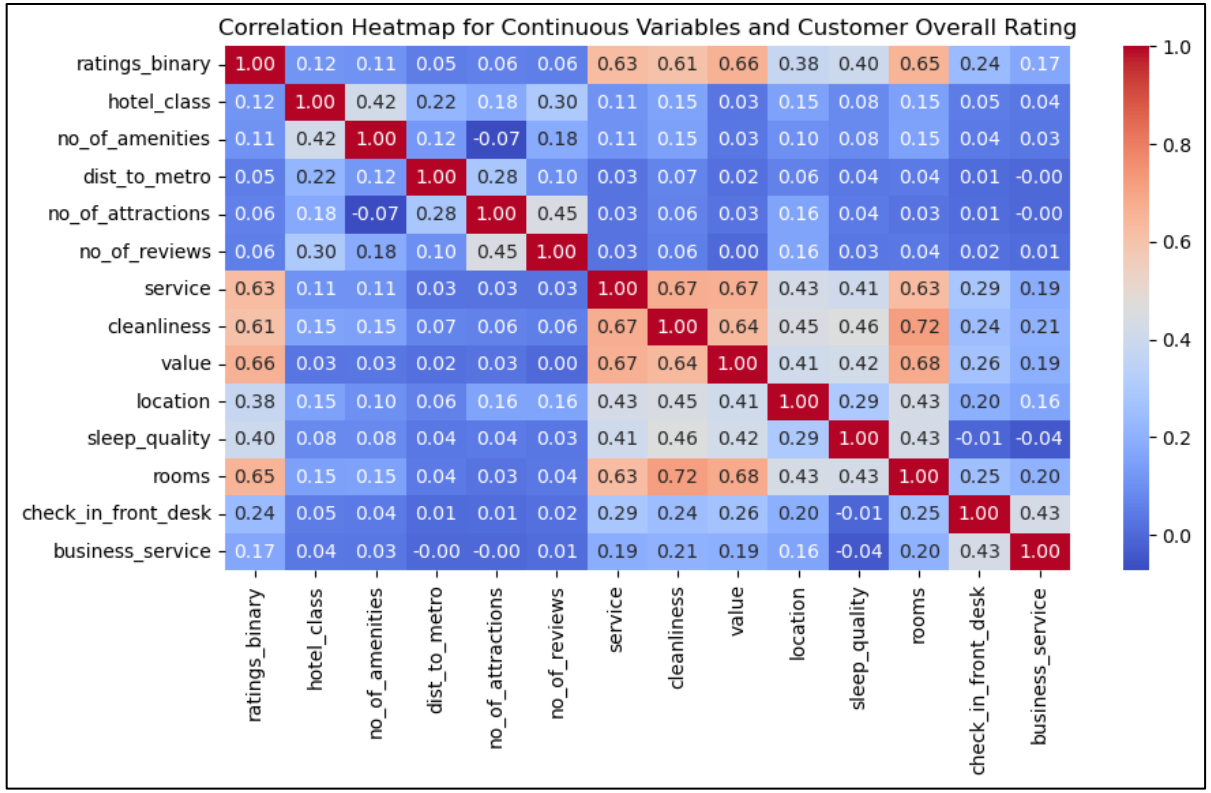
	Training Error	Validation Error	Validation Accuracy
Initially	0.208622	0.226135	0.897309
Forward Selection	0.166930	0.188483	0.921456
Backward Selection	0.168118	0.187797	0.922020

- Sequential Feature Selection (SFS) is used.
- **XGBoost consistently performed best** without SFS and with Forward and Backward Selection
- **Log loss** is chosen as the loss function because it is **sensitive to class imbalance**, and penalises wrong predictions
- Validation accuracy is calculated using accuracy score
- **Backward Selection** is the final method with lowest training and validation error, and highest validation accuracy.
- There are **14 chosen features** after SFS, most of which show high correlation with the label in the Heatmap Figure

MODELLING

FEATURE SELECTION

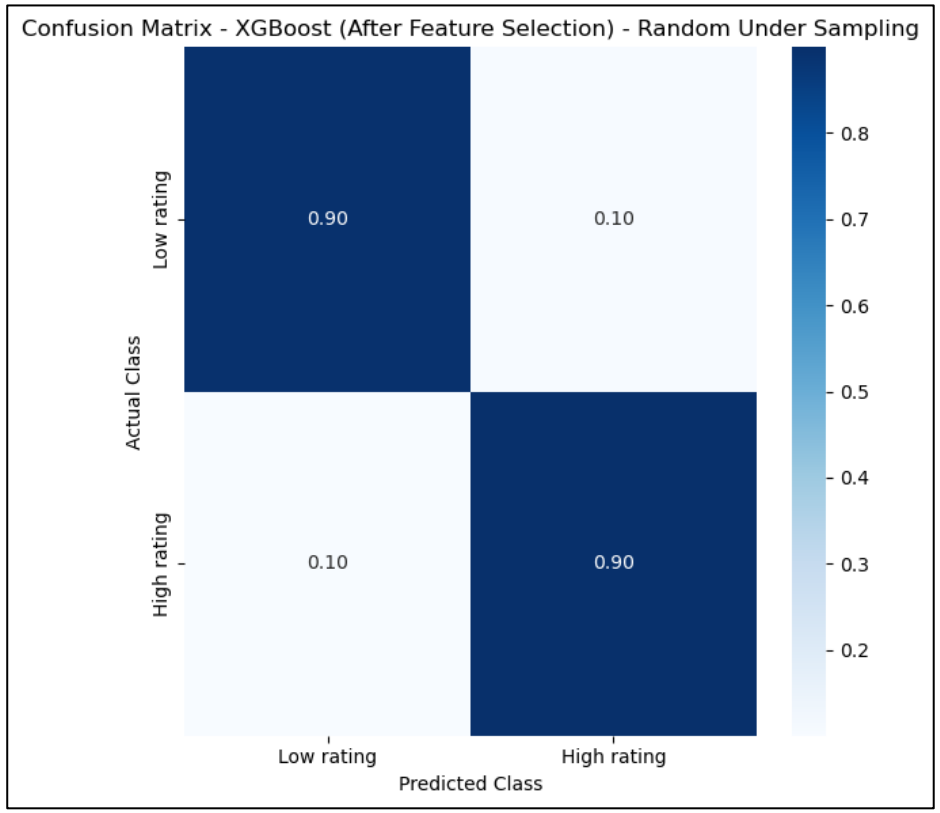


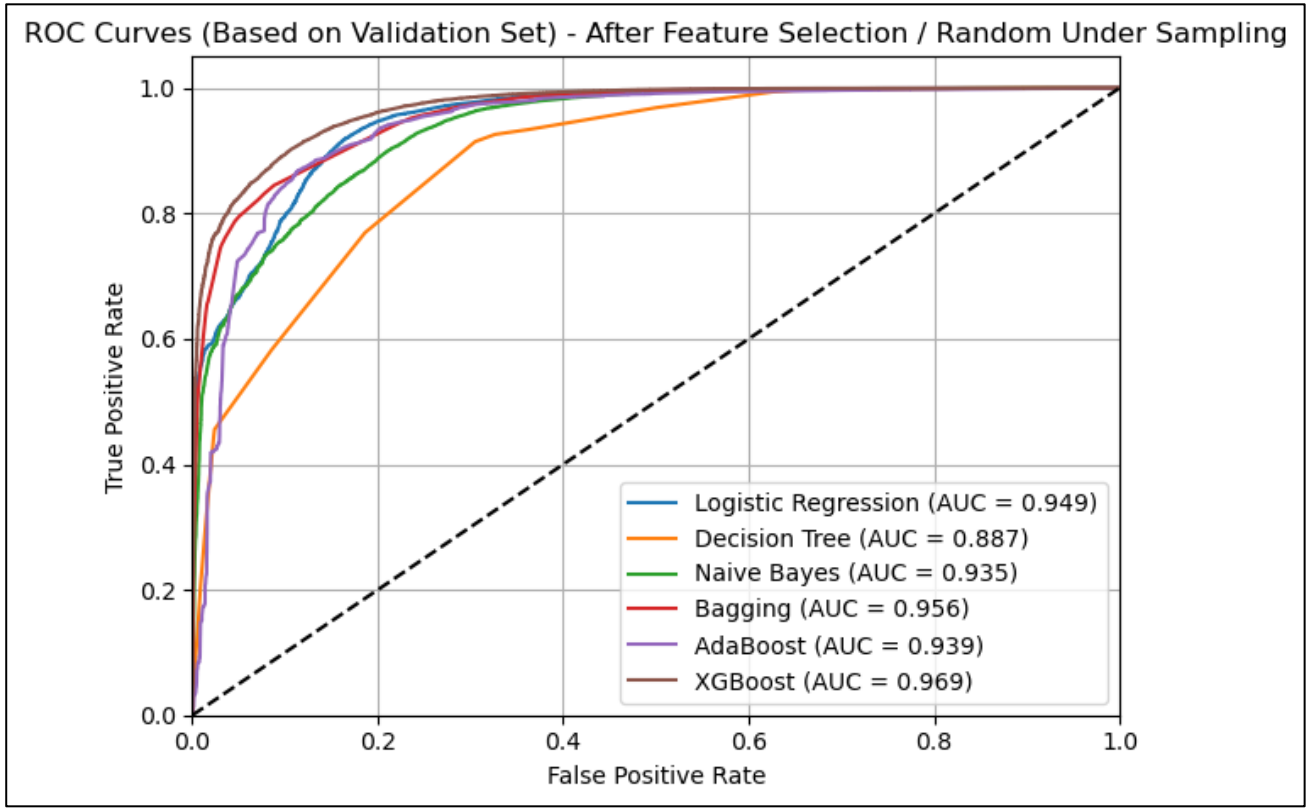


Selected features: award, no_of_reviews, service, cleanliness, value, location, sleep_quality, rooms, check_in_front_desk, business_service, sentiment, price_level_expensive, price_level_moderate, price_level_very expensive

EVALUATION

MODEL SELECTION (BASED ON VALIDATION SET)





After feature selection, when validated on validation set, **XGBoost yields the highest AUC = 0.969**, and have best normalised confusion matrix. Therefore, the final chosen model is **XGBoost with Random Under Sampling and Backward Feature Selection**.

EVALUATION

HYPERPARAMETER TUNING

Metric	Default Hyperparameters	Bayesian Search	Grid Search
Accuracy	0.8973	0.8980	0.8943
Precision	0.8970	0.8966	0.8934
Recall	0.8974	0.8996	0.9008
Specificity (TNR)	0.8971	0.8964	0.8937
F1 Score	0.8972	0.8981	0.8971

- Grid Search and Bayesian Optimization is used.
- **Bayesian has the highest AUC = 0.9698**.
- Based on the normalised confusion matrix, Bayesian Search achieves the highest figures in **accuracy, precision and F1-score**.

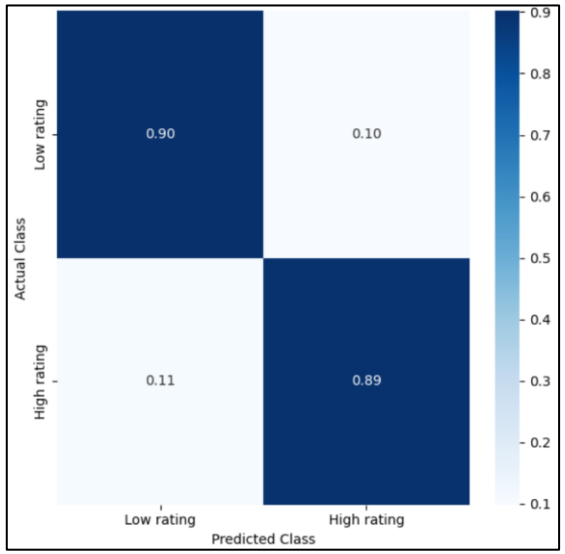
The aim of the project is to correctly predict negative ratings and minimises false positive, since Bayes Search achieves the second-best TNR – meaning it can predict negative ratings well and have high precision as well as balanced F1-score.

Bayesian Search is the final method.

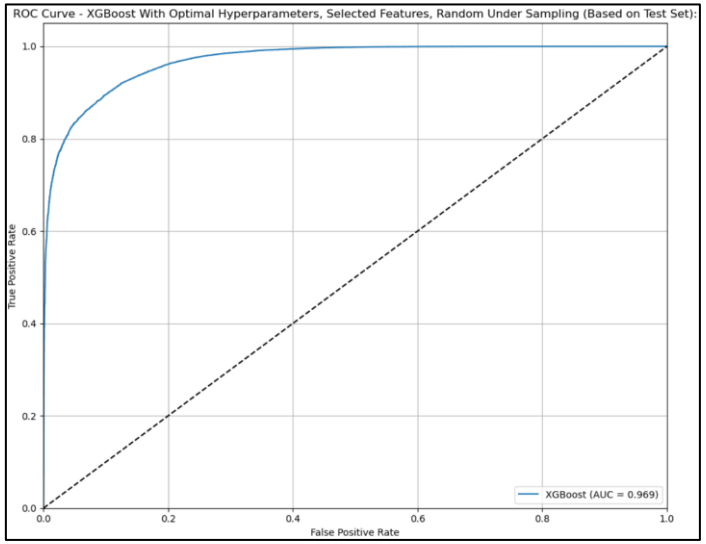
EVALUATION

FINAL MODEL SELECTION (SELECTED FEATURES & OPTIMISED HYPERPARAMETERS)

	Model	Accuracy	Precision ("low ratings" class)	Recall ("low ratings" class)	F1-score ("low ratings" class)
0	XGBoost (Optimal Hyperparameters)	0.896099	0.743861	0.901809	0.815255



- This test was **based on the test set**.
- **It is of utmost importance for the hotel to correctly predict the dissatisfied customers (low ratings)**, so that actions can be taken to prevent potential customer churns. Failing to recognise these customers can be more costly than mistakenly targeting satisfied ones for enhancements or compensation. Especially in the highly competitive hospitality industry, negative reviews can severely damage a hotel's reputation and public image.
- This is why we selected the final model **based on the highest recall**, which reflects the model's ability to correctly predict True Negative.
- The chosen model is the **XGBoost** classifier, trained on a balanced dataset using **Random Under Sampling** and selected features from **Backward Selection** (14 features). With optimised parameters obtained from **Bayesian Search**, the model achieved an overall accuracy of **0.896** and a recall rate of **0.902** for the 'low rating' class.



BUSINESS INSIGHTS AND RECOMMENDATIONS

With optimised parameters from Bayesian Search, XGBoost trained on a balanced dataset using Random Under Sampling and selected features from Backward Selection achieved the highest recall rate of 0.902 for the 'low' rating class. Hotels are recommended make use of this model that is proven helpful for decision-making.

THE MODEL OF CORRECT PREDICTION OF LOW RATINGS ARE USEFUL FOR HOTELS IN DIFFERENT WAYS.

- **Early identification of service failures**: If many negative reviews mention a certain key word (cleanliness, staff behaviour, room quality etc.), it is a signal for improvement. Hotels can quickly identify recurring service failures or other bottlenecks.
- **Timely intervention**: Predicting low ratings allows hotels to engage dissatisfied guests proactively and resolve their issues before they check out, increasing the likelihood of repeat bookings and positive word-of-mouth. Addressing concerns during or immediately after a guest's stay by offering solutions such as room upgrades, discounts, or personalised attention can mitigate any adverse effects and prevent negative reviews going viral.

FURTHER MANAGERIAL IMPLICATIONS CAN ALSO BE MADE FROM THE MODEL IN GENERAL.

- **Competitive benchmarking**: Analysing behaviours of guests helps hotels gain various insights into competitor performance and strategies and can thus adjust their plans accordingly.
- **Product or service development**: Tracking sentiment trends over time provides a good general picture of whether new products or services are being well received.
- **Operational efficiency**: Recurring key words in either positive or negative reviews highlight problems with specific areas of hotel operations, enabling management to direct training or resource adjustments where they are most needed.

Reducing revenue losses from poor retention and staying competitive in the hospitality market.