Hotel Review Sentiment Analysis & Rating Prediction

DSC511: Big Data Analytics
Maria Tsilidou, Anastasios Nikodimou, Ioannis Demetriou

Overview

- Introduction
- Dataset Description and Cleaning
- Exploratory Data Analysis
- Natural Language Processing
- Graph Analysis
- Machine Learning
- Limitations
- Conclusion

Introduction & Motivation

The Digital World of Hotel Booking

- Today, most people read online reviews before choosing a hotel.
- Websites like **Booking.com** are full of real experiences from travelers.

Analyzing Reviews Can...

- Identify common issues across different hotels
- Help hotels improve their services based on real guest experiences

Project Objectives

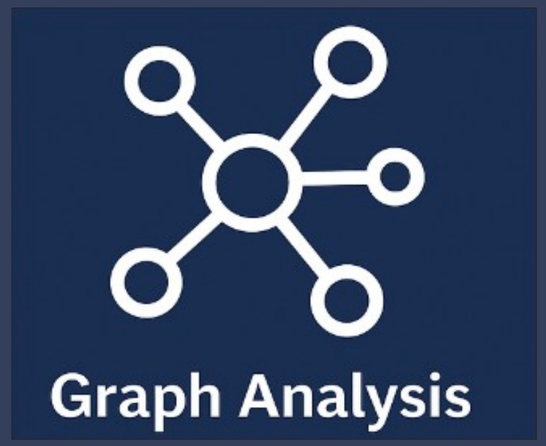
- **Understand review patterns**:
  Analyze how guests express opinions in positive and negative terms.
- **Predict user satisfaction**:
  Use machine learning to predict the overall hotel rating based on review text.

Dataset Description

Source: Booking.com
Size: over 515,000 reviews
Hotels Covered: 1,493 hotels across Europe
Fields Included: 17 variables

Variable Name	Description
Hotel_Address	
Review_Date	
Average_Score	Hotel’s average rating.
Hotel_Name	
Reviewer_Nationality	Country of origin of the reviewer.
Negative_Review	
Review_Total_Negative_Word_Counts	
Positive_Review	
Review_Total_Positive_Word_Counts	
Reviewer_Score	The score the reviewer gave to the hotel.
Total_Number_of_Reviews_Reviewer_Has_Given	Total reviews written by this reviewer.
Total_Number_of_Reviews	Total number of reviews the hotel has received.
Tags	Descriptive tags (e.g., "Business trip", "Solo traveler").
days_since_review	Days between the review date and the data collection date.
Additional_Number_of_Scoring	Number of score-only ratings (no text review) for the hotel.

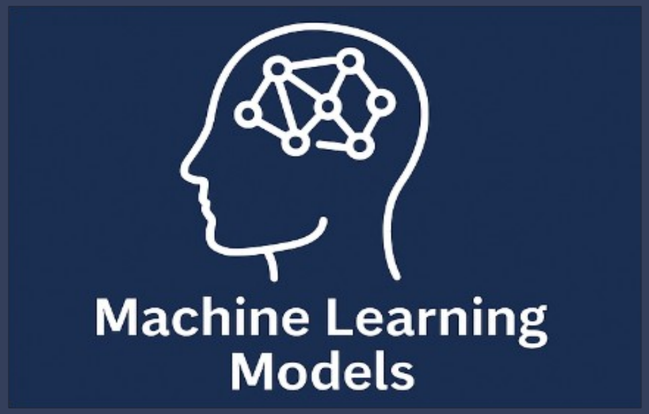
What We Used in Our Project



We used graph-based techniques to explore relationships between reviewers and hotels (e.g., shared nationalities or common reviews), identifying patterns and influential nodes.

We built a system that suggests hotels to users using the graph analysis. For each hotel and customer, we generated ids. Those ids were used to identify a unique customer and which

We analyzed positive and negative reviews by cleaning, tokenizing, and extracting features like word counts or sentiment, helping us understand what guests liked or disliked.

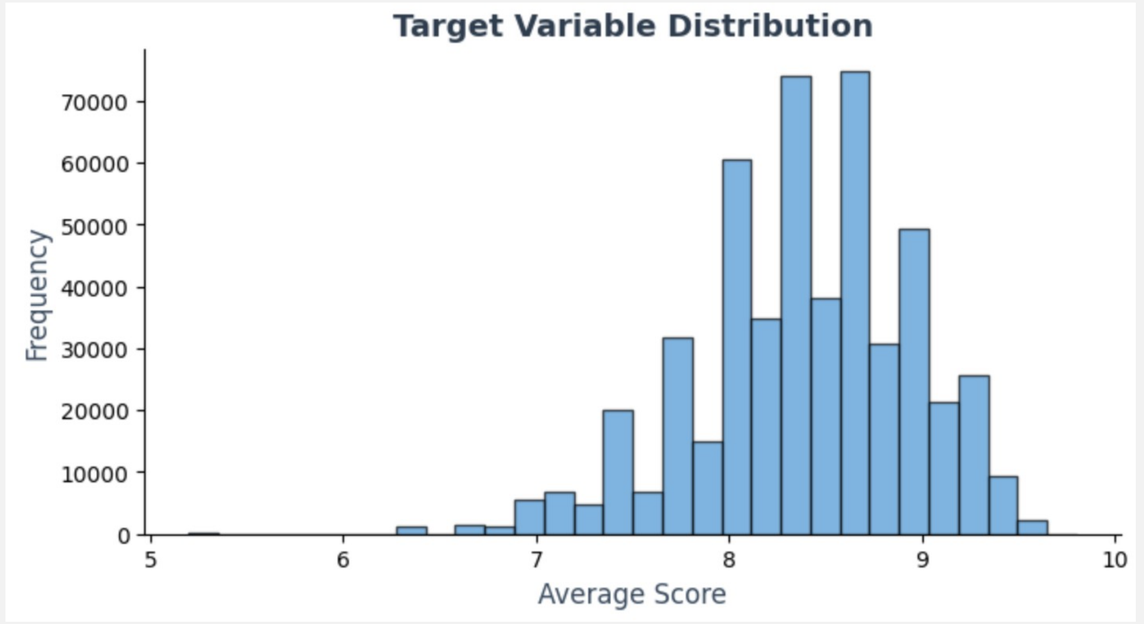


We trained models to predic review sentiment or hotel satisfaction using structured features and text, improving our understanding of what influences guest ratings.

Dataset Description and Cleaning

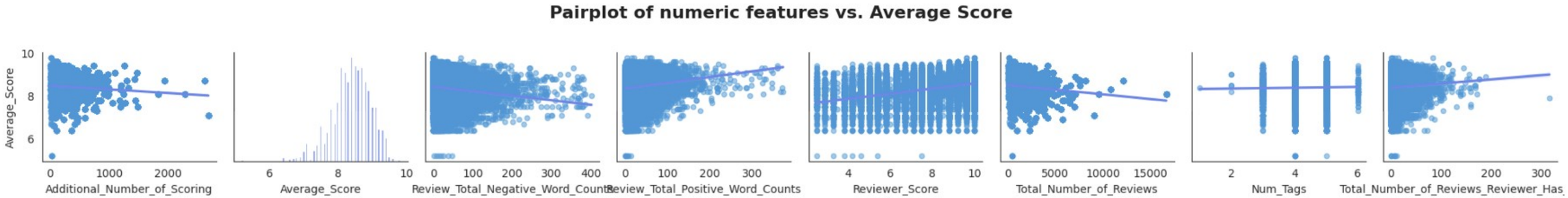
- Poorly encoded missing values (empty characters, ‘none’, ‘NA’ etc).
- ‘No Positive’ and ‘No Negative’ in described the absence of positive or negative review, respectively.
- Duplicated values were found.
- Longitude and latitude represented as strings instead of numeric. They also contained missing values.
- Time related features were typed as string

Exploratory Data Analysis



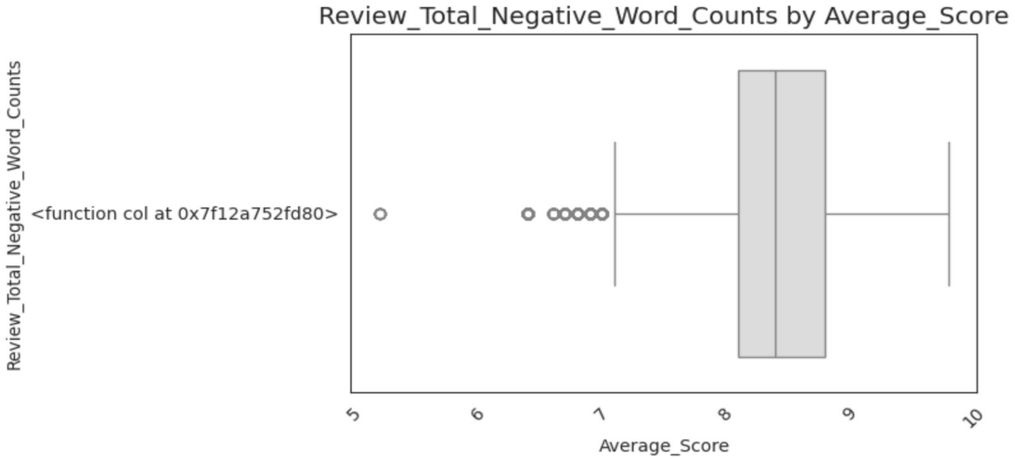
- Visualizing a sample of dataset
- Average hotel reviews are left skewed
- Concentrated around 8.4
- Investigation showed hotels in dataset were luxury

Exploratory Data Analysis



- Individual reviewer review `Reviewer_Score`, and the total "positive" words used in the review `Review_Total_Positive_Word_Counts` is positively correlated with `Average_Score`.
- The more reviews a user has given Total_Number_of_Reviews_Reviewer_Has_Given is also positively correlated with the venue's average score.
- We expected that Review_Total_Negative_Word_Counts negatively correlates to a venue's average score.
- The more reviews a venue has Total_Number_of_Reviews, the smaller its average score.
- Presence of linear relationship is an suggest that linear models will perform well.

Exploratory Data Analysis

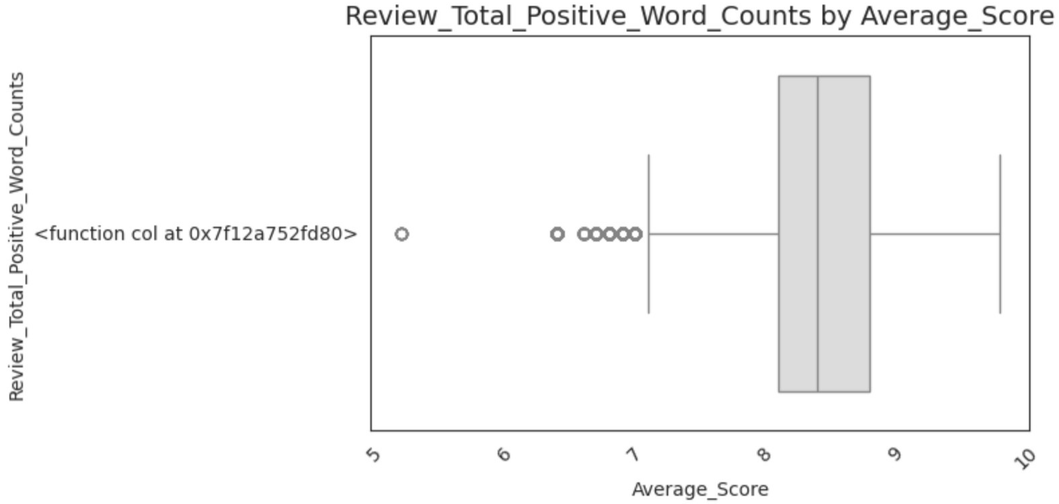


Review_Total_Negative_Word_Counts by Average_Score:

- There’s a slight downward trend: as Average_Score increases, the number of negative words tends to decrease.
- The median is generally lower for higher scores.
- The distribution is very wide at each score — especially for mid-range scores (e.g., 7.0 to 8.5), meaning high variability in how many negative words are used regardless of rating.
- Many outliers, especially in lower scores, suggest some extreme reviews with very high negativity.

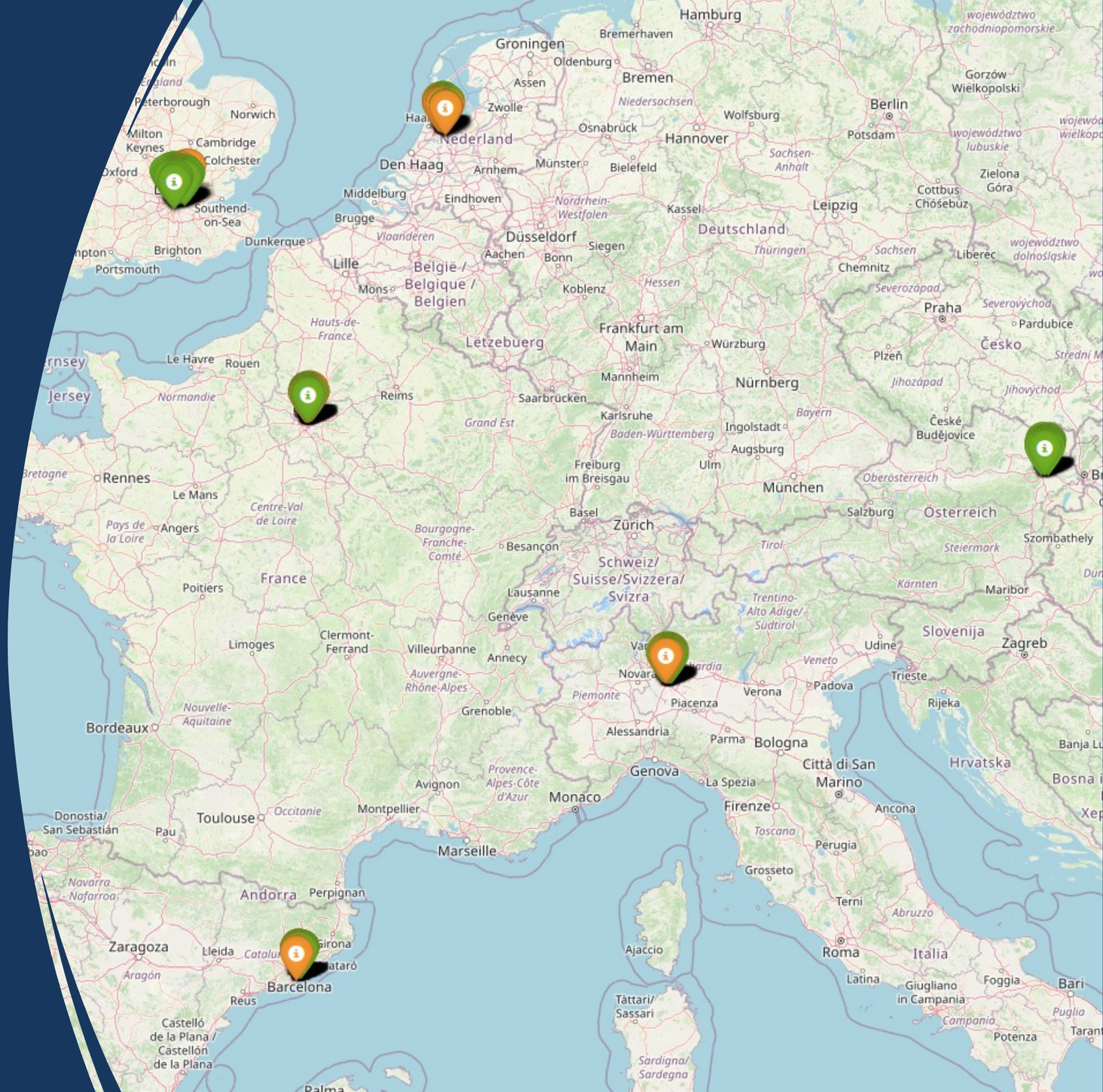
Review_Total_Positive_Word_Counts by Average_Score:

- Positive trend: as Average_Score increases, the number of positive words also increases.
- The median positive word count rises steadily from low to high scores.
- Whiskers and outliers increase with higher scores — people who are more satisfied tend to say more and more positive things.



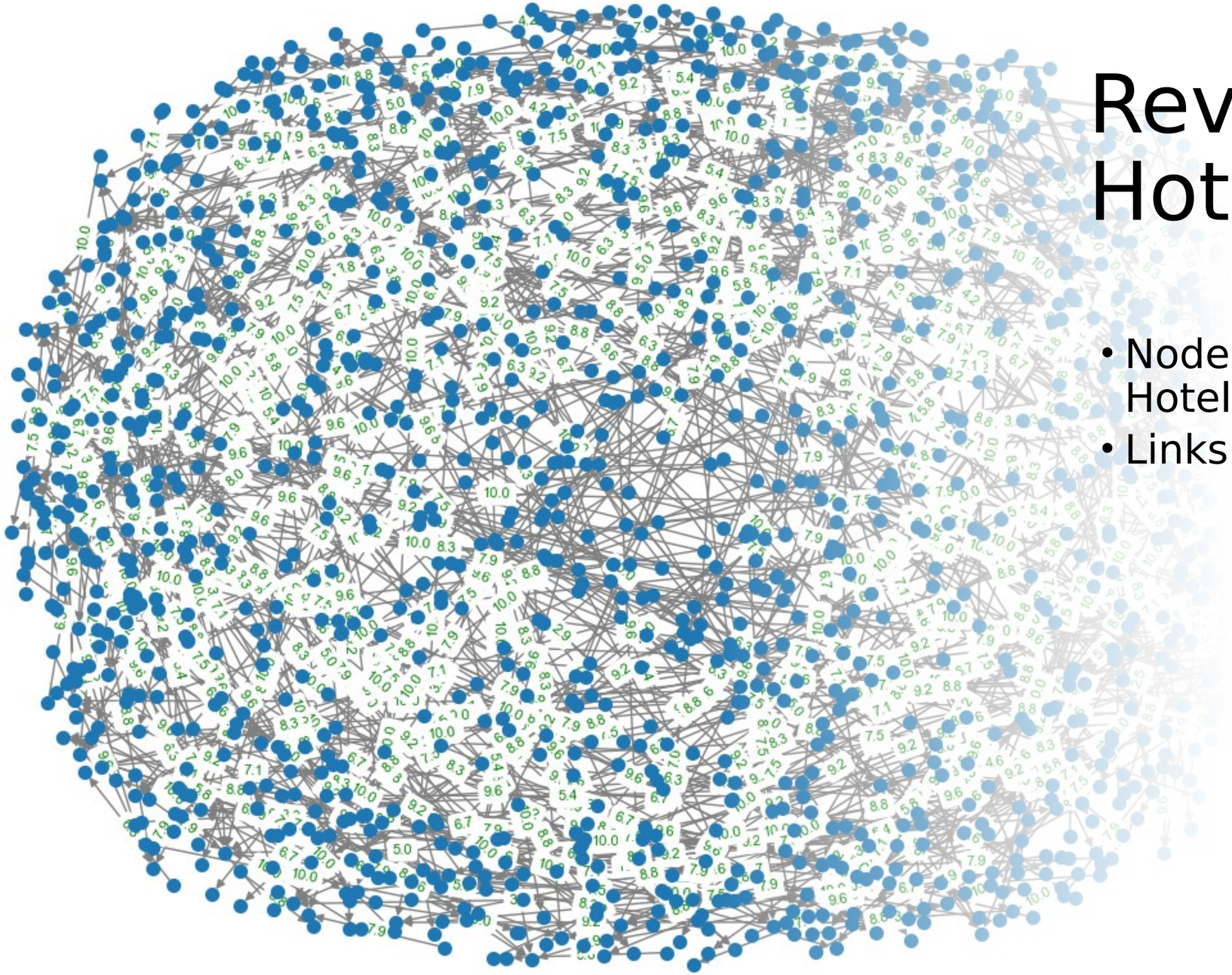
Geographical Distibution

- Vienna, Austria
- Paris, France
- Amsterdam, Netherlands
- Barcelona, Spain
- Milan, Paris
- London, United Kingdom



Graph Analysis

- Reviewer ID was not provider
- Engineered reviewer identifier using
  - **Reviewer_Nationality:** which gives geographic context of the reviewer.
  - **Tags:** reflects the type of traveler (e.g., 'Solo', 'Couple').
  - **Total_Number_of_Reviews_Reviewer_Has_Given:** adds some numeric uniqueness.



Reviewer - Hotel

- Nodes: Reviewer and Hotels
- Links: Ratings

Interpretation of the graph

- Dataset large Graph not interpretable
- We created a code to manually rank the 10 most reviewed hotels
- Some of those hotels were 'Strand Palace Hotel' (2525) with average score 8.1, 'Copthorne Tara Hotel London Kensington' (2231)with average score 8.1, 'Britannia International Hotel Canary Wharf' (2112) wiha average score 7.1, 'Hotel Da Vinci' (1783) with avrage score 8.1, 'Millennium Gloucester Hotel London' (1650) with average sc ore 7.8

Collaborative Filtering

- Reviewer Identifier (Engineered) - Hotel Name
- Alternative Least Squares (ALS)
- 80-20 Train / Validation Split

Collaborative Filtering

reviewer_id_index	hotel_id_index	rating
12	4	8.33
12	712	5.44
12	101	5.23
12	123	5.02
12	91	4.97
12	376	4.73
12	1251	4.71
12	445	4.68
12	36	4.65
12	527	4.44

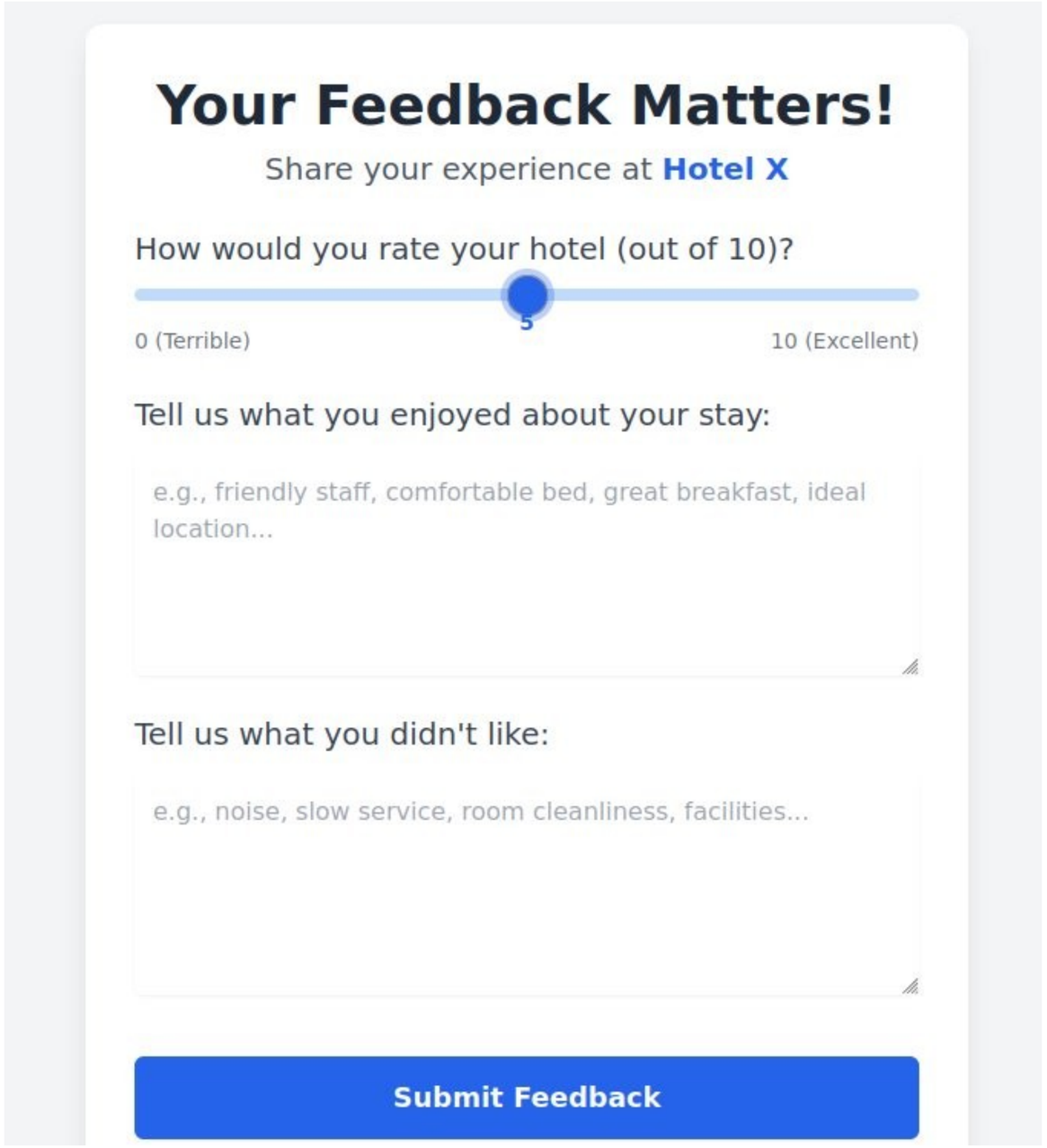
Root Mean Square Error = 7.15

Natural Language Processing

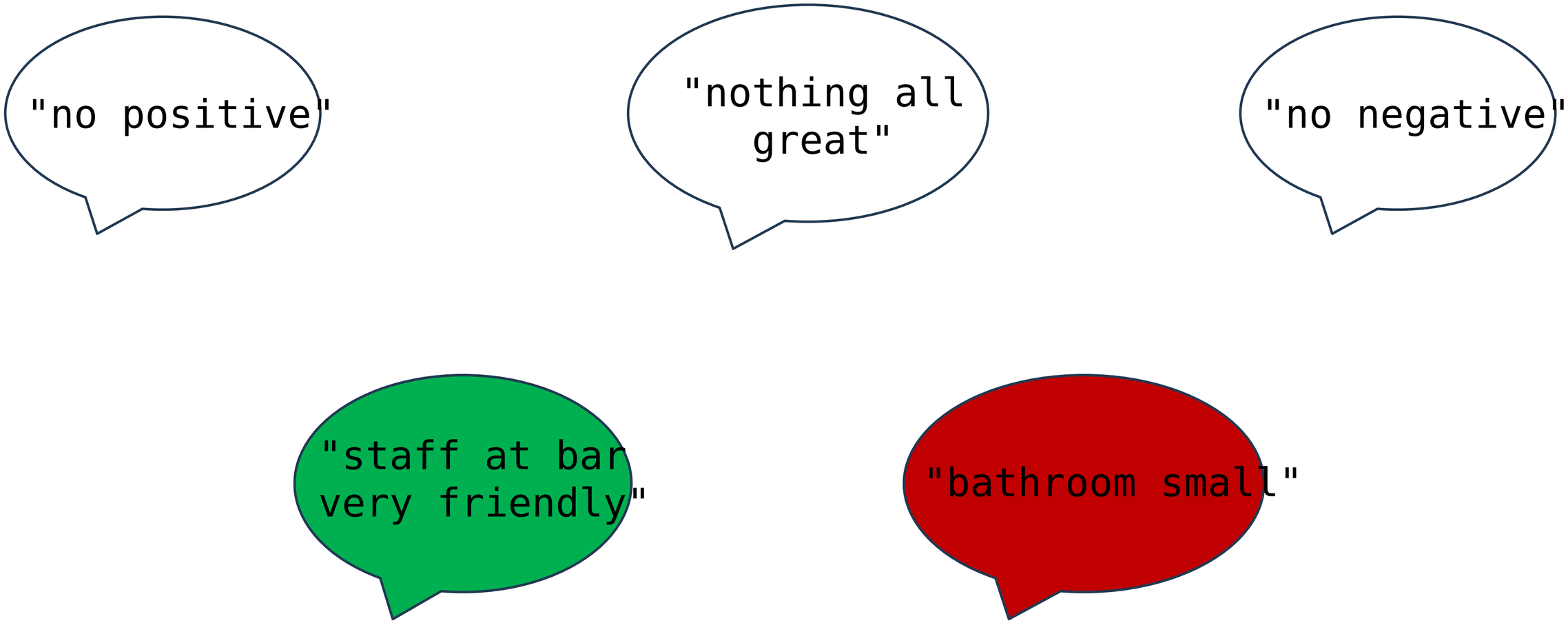
Reviews split in "Negative" and "Positive"

Can use as labels for sentiment analysis

Natural Language Processing



Natural Language Processing



Natural Language Processing

Tokenization

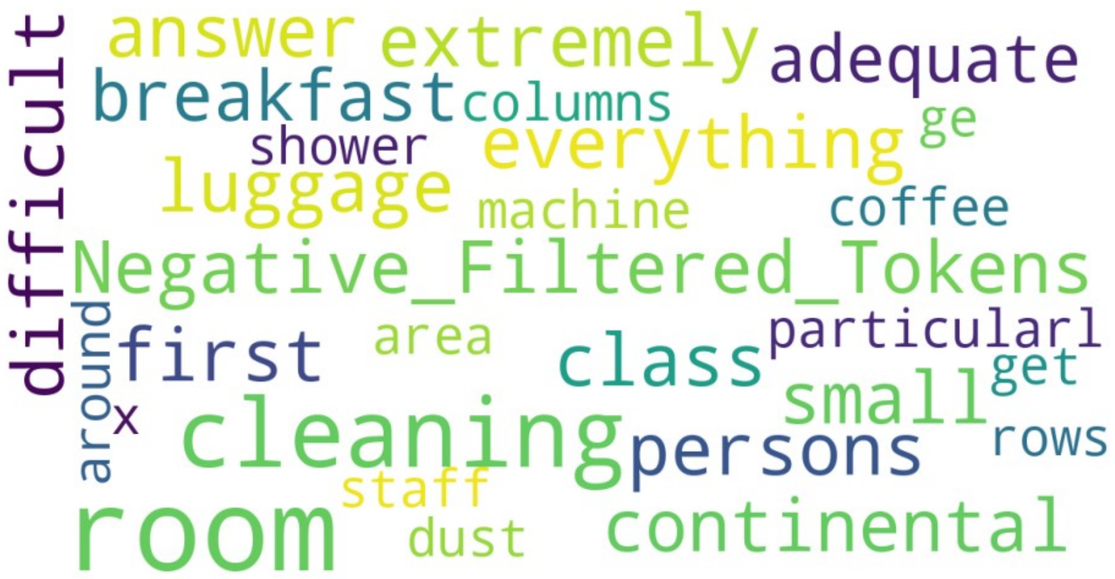
Text Cleanup

Stop Word Removal

Lemmatization (nltk)

Natural Language Processing

Negative Tokens



Positive Tokens

Machine Learning Techniques

Linear Regression Model

- Generated an aggregations for the numeric columns over each hotel in the dataset to calculate the averages.
- Cross-validation was not used during this process.
- Working from a sample by performing selective sampling and keeping only the hotels with the most reviews (48858 / 515212).
- Building a regression model to predict the average hotel review score based on aggregated features derived from user reviews.
- Taking a sample from the aggergated dataset (1415 / 515212).
- We then filter out hotels with fewer than 30 reviews and those missing geographic data.
- Evaluation: Using RMSE and R^2
- RMSE on test set: 0.3 and R^2 on test set: 0.49
- The results indicate a moderate predictive performance could potentially be improved by using more complex models (like Random Forest or Gradient Boosting)

GBT and Random Forest evaluation

Results:

Linear Regression - RMSE: 0.3776, R^2: 0.4913

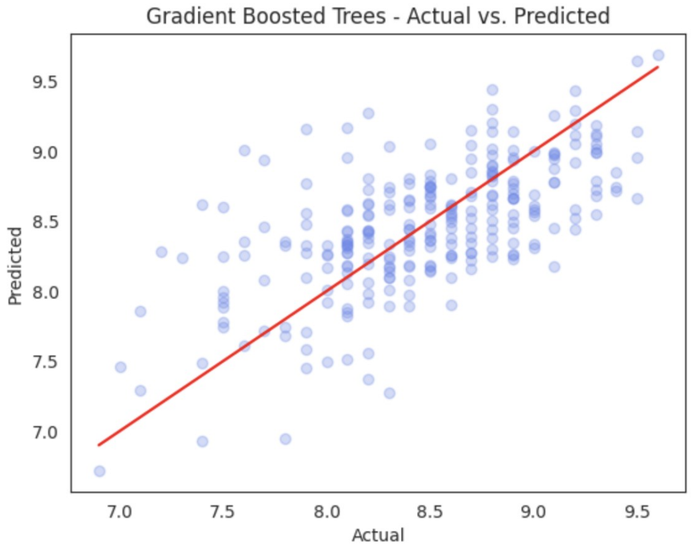
Random Forest - RMSE: 0.3810, R^2: 0.4820

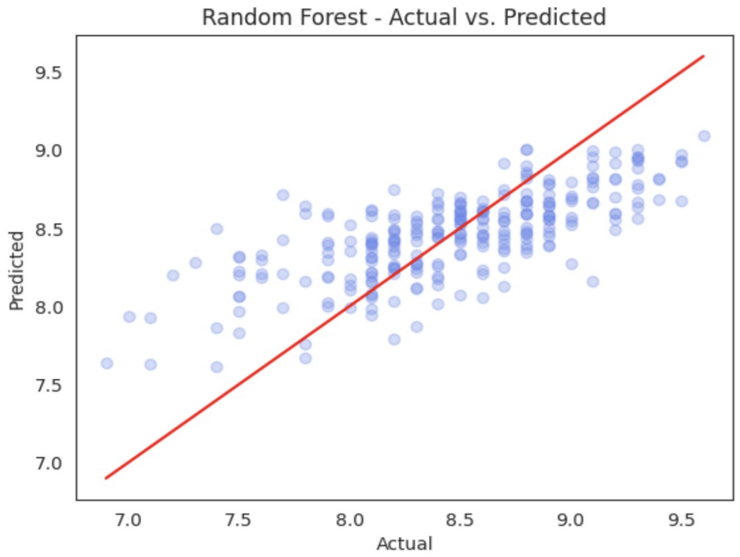
Gradient Boosted Trees - RMSE: 0.4255, R^2: 0.3539

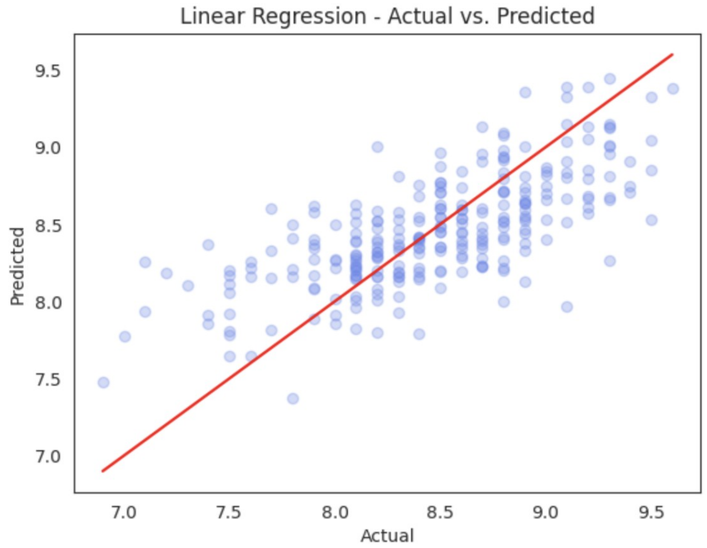
⇒ Linear Regression performs best among the three models

⇒ LR captures the underlying patterns in your aggregated dataset more effectively, so this implies that our data relationships are relatively linear, or at least well-approximated by a linear function at this aggregation level.

Scatter plots of actual vs the predicted values







MODEL 3 Sentiment Classification Approach

Model 3: Sentiment Classification Approach

- We separated the positive and negative parts of each hotel review.
- Each part was treated as a separate training example:
- Positive comments → **Label 1**
- Negative comments → **Label 0**
- This helped us to :

isolate clear sentiments by separating the positive and negative parts of the reviews.

- We converted text to numbers using **HashingTF** and **IDF**
- Trained a **Logistic Regression** model to predict sentiment

Model 3: Sentiment Classification Approach

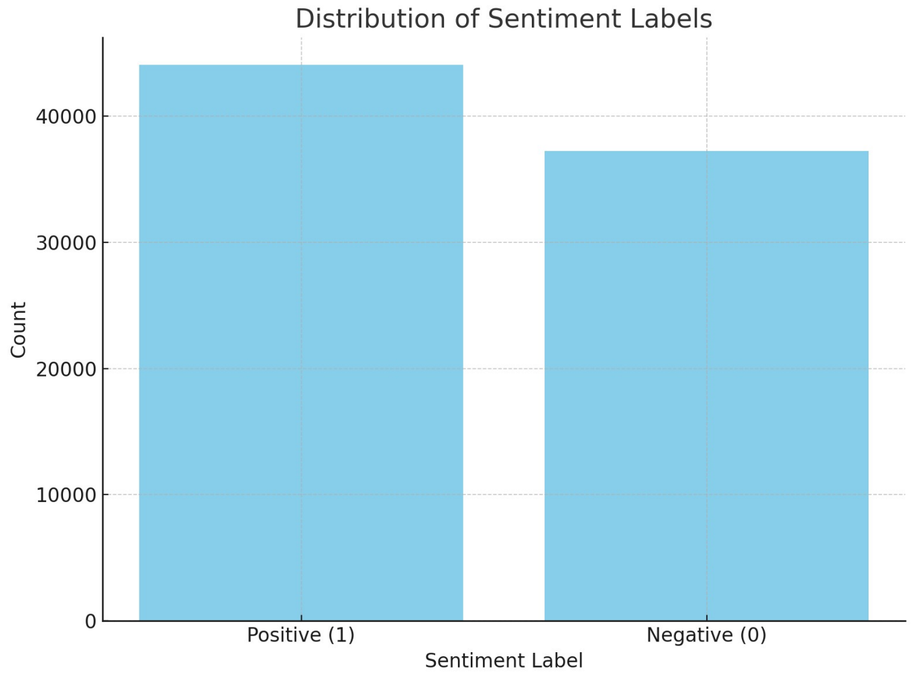
In this step, we are analyzing the distribution of sentiment labels within our dataset. Each review is labeled as either:

- **1 (Positive sentiment)**
- **0 (Negative sentiment)**

- The dataset is **not imbalanced**, which means we can proceed with training our classification models without applying aggressive resampling techniques.

Process:

- We split the data into **80% training** and **20% testing**
- We trained a logistic regression model on the training set



Model 3: Sentiment Classification Approach

Metric	Value	Interpretation
Accuracy	0.8888	About 89% of all reviews were correctly classified.
F1 Score	0.8888	A balanced measure of precision and recall – indicates reliable predictions.
Weighted Precision	0.8888	Our model predicts a sentiment, it's correct ~89% of the time.

Conclusion:
Logistic regression model performs reliably across both positive and negative sentiment classes, with no major imbalance or bias.

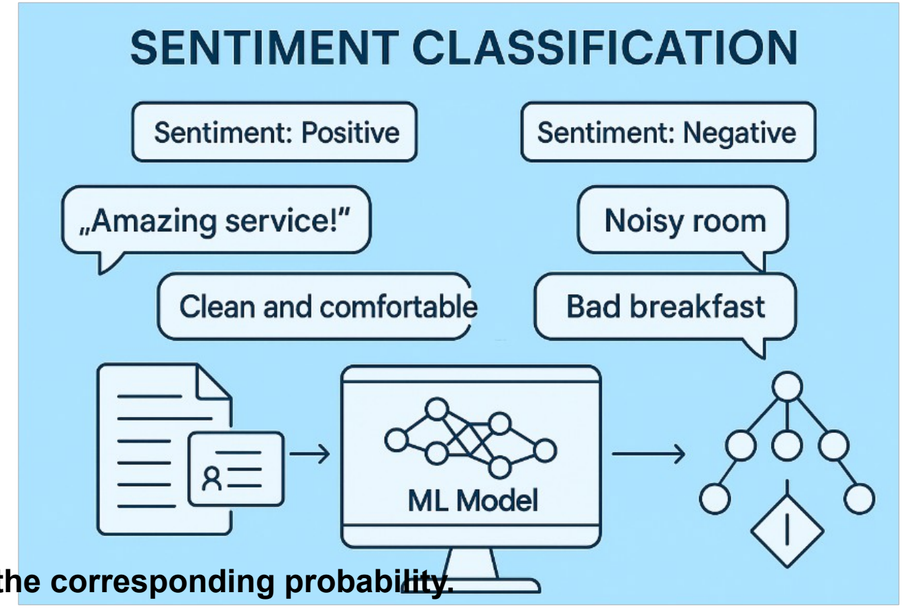
After predicting review sentiment, we compared predictions to true labels to understand model performance using a confusion matrix.

Confusion Matrix

	Predicted Positive (1)	Predicted Negative (0)
Actual Positive (1)	True Positive (TP): 7992	False Negative (FN): 886
Actual Negative (0)	False Positive (FP): 926	True Negative (TN): 6494

Classifying New Hotel Reviews Using a Trained ML Model

We use our trained machine learning pipeline to **predict whether new hotel reviews are positive or negative** based on their text.



Input: New Example Reviews
We manually entered a few hotel reviews, both positive and negative, to test the model's performance on unseen data.
Examples:
•Positive: "*The room was spotless and the staff were incredibly friendly.*"
•Negative: "*There was no hot water and the heater was broken.*"

ROOM 01

We:
•Converted these reviews into a Spark DataFrame.
•Used the trained pipeline to classify them

Extracted the prediction (1.0 for positive, 0.0 for negative) and the corresponding probability.

$ 2.900,56

Review Text	Prediction	Probability (Negative, Positive)
The room was spotless and the staff were incredibly friendly.	1.0	[0.1792, 0.8208]
There was no hot water and the heater was broken.	0.0	[0.8024, 0.1976]

Classifying New Hotel Reviews Using a Trained ML Model

- We use our trained machine learning pipeline to predict whether new hotel reviews are positive or negative based on their text.
- We manually entered a few hotel reviews, both positive and negative, to test the model's performance on unseen data.

Examples:

- Positive: "*The room was spotless and the staff were incredibly friendly.*"
- Negative: "*There was no hot water and the heater was broken.*"

Extracted the prediction (1.0 for positive, 0.0 for negative) and the corresponding probability.

Review Text	Prediction	Probability (Negative, Positive)
The room was spotless and the staff were incredibly friendly.	1.0	[0.1792, 0.8208]
There was no hot water and the heater was broken.	0.0	[0.8024, 0.1976]

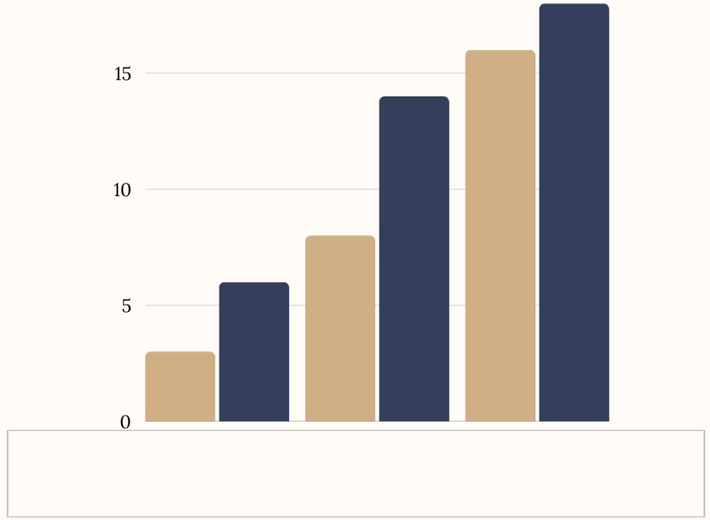
Sentiment Classification Model Applications

Customer Feedback Monitoring

•Automatically classify reviews as positive or negative.
•Track customer satisfaction trends over time.

Real-Time Alert Streaming System

- Trigger alerts when negative sentiment crosses a threshold (e.g., many bad reviews in a short period).
- Useful for hotels to respond quickly to service issues, crises.



Model 4: Hotel Star Rating Prediction using Review

In this part of the project, we shift from analyzing raw review text to working with **aggregated numerical features** derived from customer feedback. These features include:

- Average review score
- Number of review tags
- Count of positive and negative lemmas
- Total number of reviews per hotel

- To simplify the prediction task, we created a new target variable called star rating, which maps each hotel's average review score to a 1-to-5-star scale.

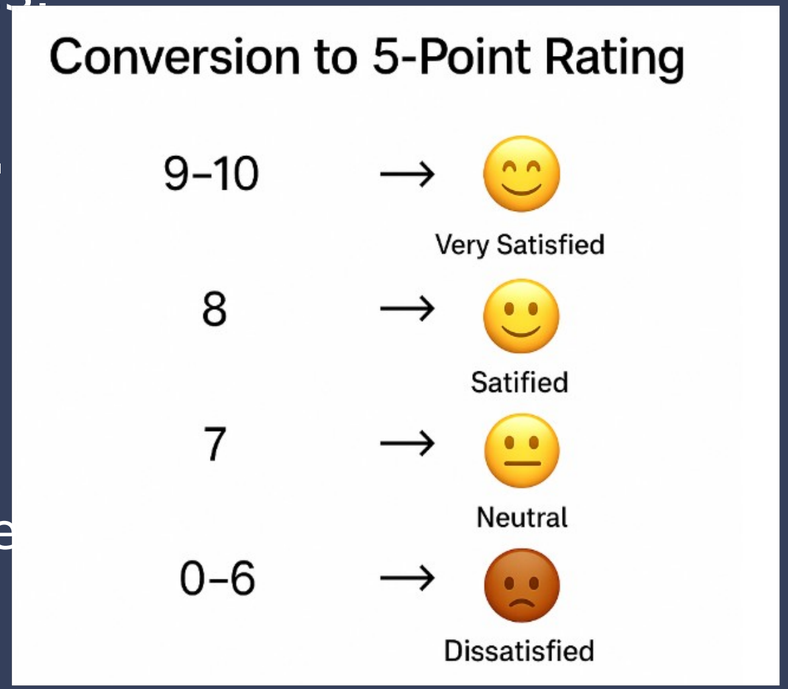
This transformation allows us to approach the problem as a multi-class classification task, making it easier to evaluate model performance across distinct hotel quality levels.

Random Forest classifier (initial approach)

One of the strategies we applied was transforming the original hotel score from a 0–10 scale into a 1–5 star rating.

This allowed us to reframe the problem as a classification task and evaluate how well our model performs in predicting the perceived star quality of hotels based on their average customer re ew scores.

We could have changed the 0–10 scores into **satisfaction levels.**



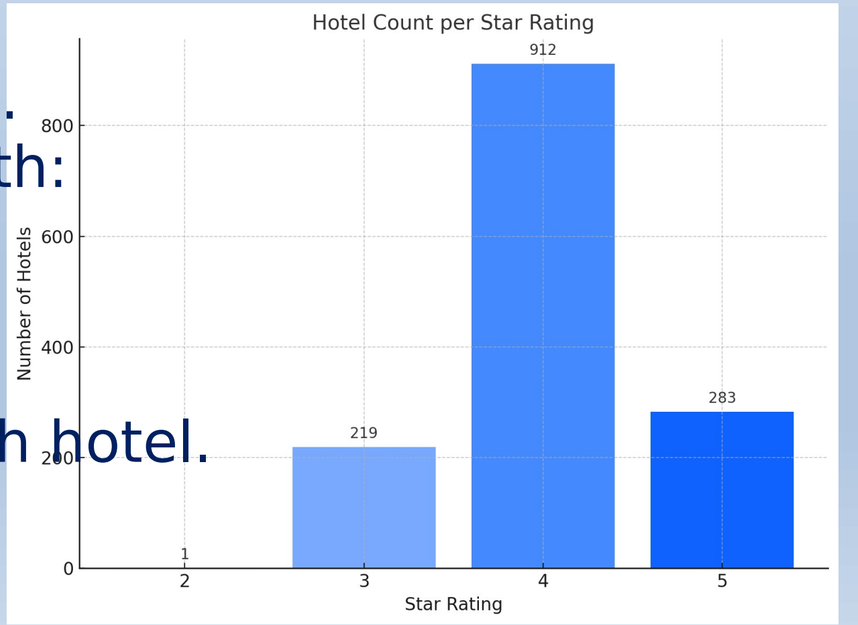
For example, instead of using a star rating, we could classify scores into labels
such as *Dissatisfied* and *Satisfied*, which might better capture the emotional
tone behind the reviews and offer more interpretable insights for business decision-making.

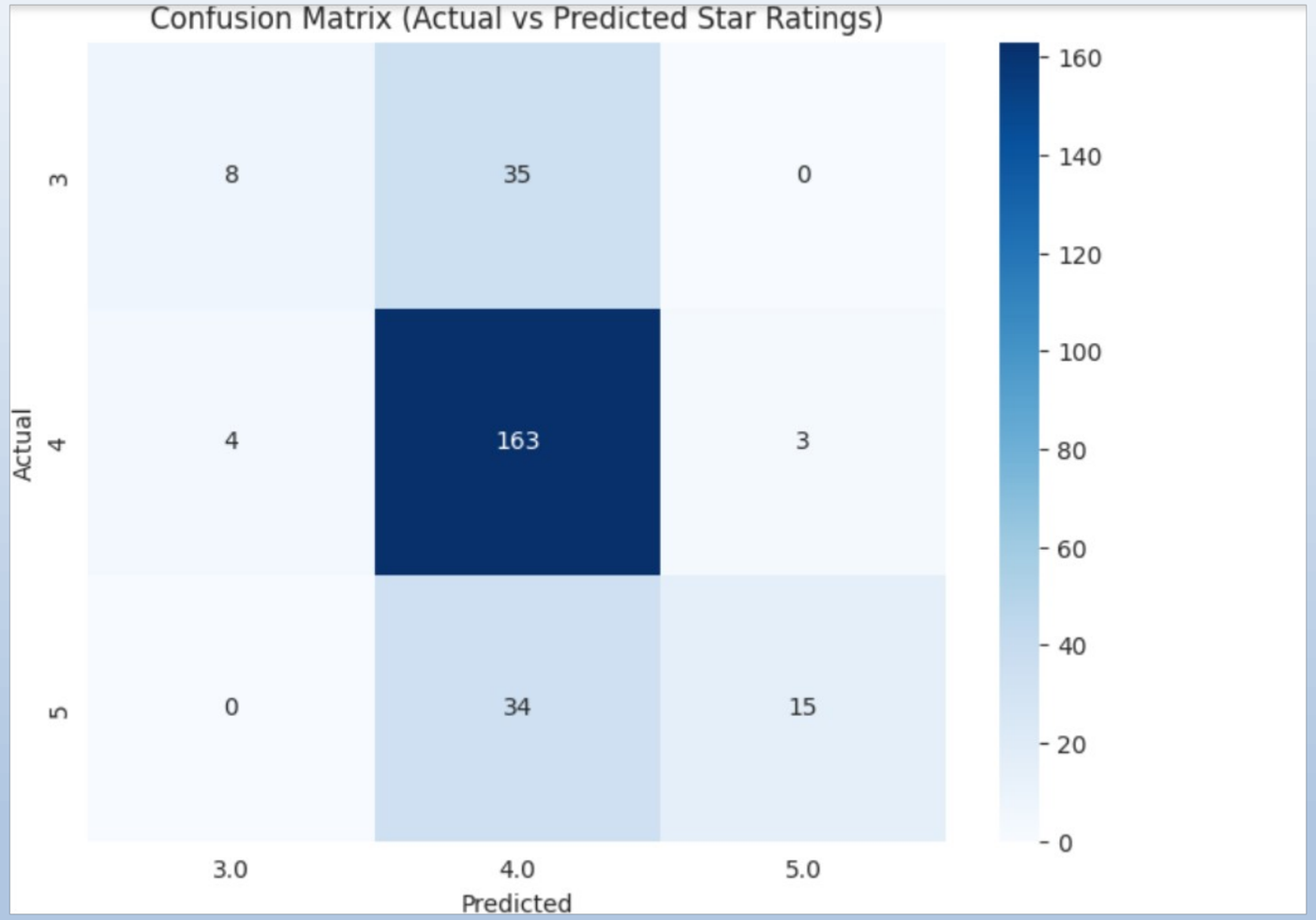
Random Forest Classifier Results

- Before training our model, we analyzed the distribution of hotel star ratings in the dataset.
- We included only 3,4 and 5-star hotels.

- We built a Random Forest classifier to predict the star rating using
- these features.
- Split the dataset into training and test sets (80/20 split).
- Trained the model using a machine learning pipeline with:
- A Vector Assembler to combine features
- A Random Forest Classifier to learn patterns Generated predictions

and compared the actual vs predicted star ratings for each hotel.





Confusion Matrix Insights

•The majority of **4-star hotels** were correctly classified (163 out of 170).

•**3-star hotels** were often confused with 4-star ones (35 misclassified).

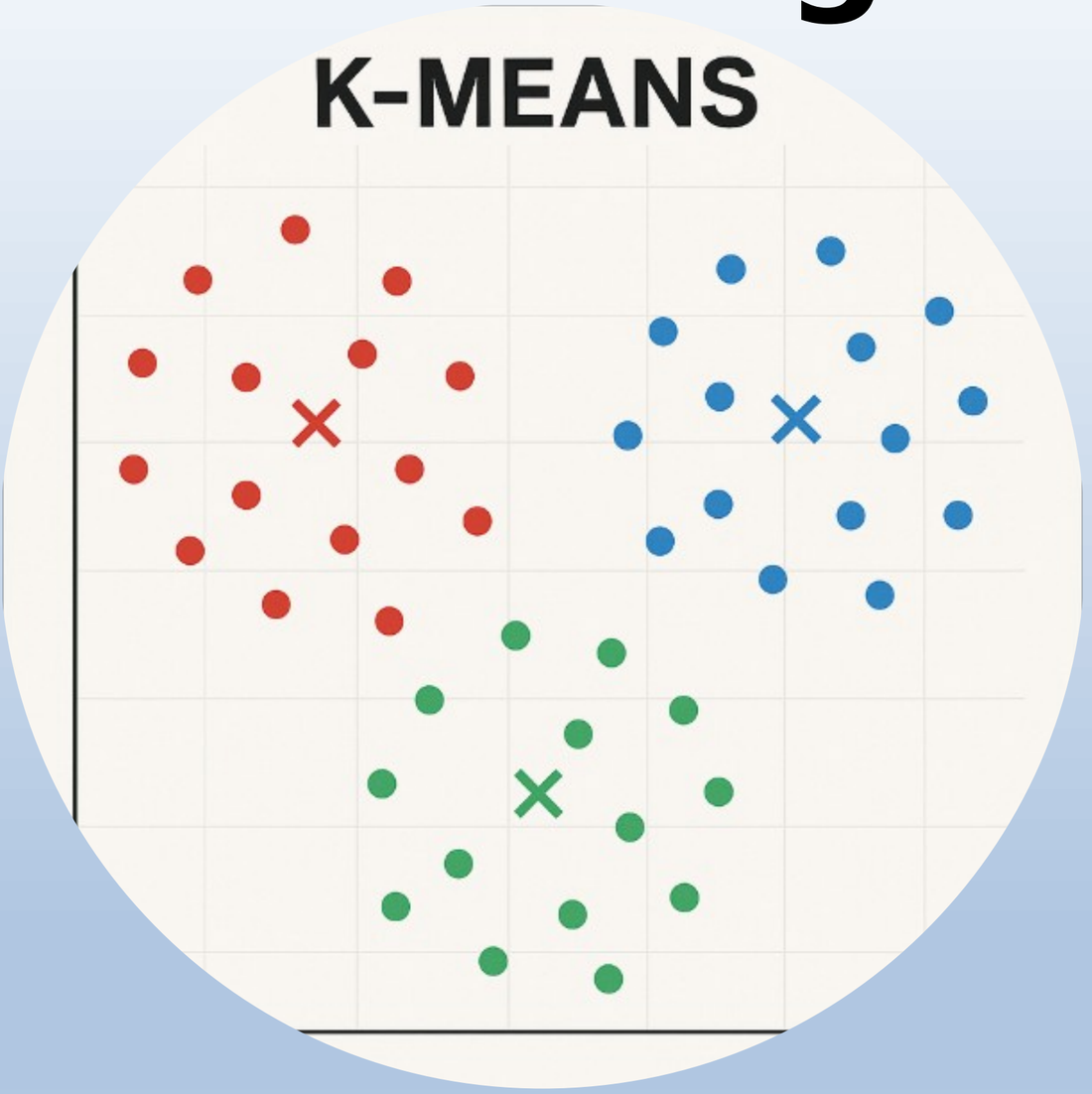
•**5-star hotels** were also frequently predicted as 4 stars (34 misclassified).

This indicates that our model is **biased toward the 4-star class**, which is expected given the class imbalance.

Future improvements could involve **resampling techniques** or **class weighting** to handle the imbalance.

Metric	Score
Accuracy	0.7099
F1 Score	0.6577

Clustering



We used these features :
Average review score
Number of positive and negative words
Number of review tagsTotal
Number of reviewsDays

We choosed the Best Number of Clusters (k)

•A good k shows clear patterns in the data
What we did:
•Tested k from 2 to 8
•Calculated WSS (how tightly hotels fit in the group)
What we found:
•WSS drops fast until k = 4, then slows down
•This means k = 4 is a good choice
→ Balances simplicity and accuracy

Clustering

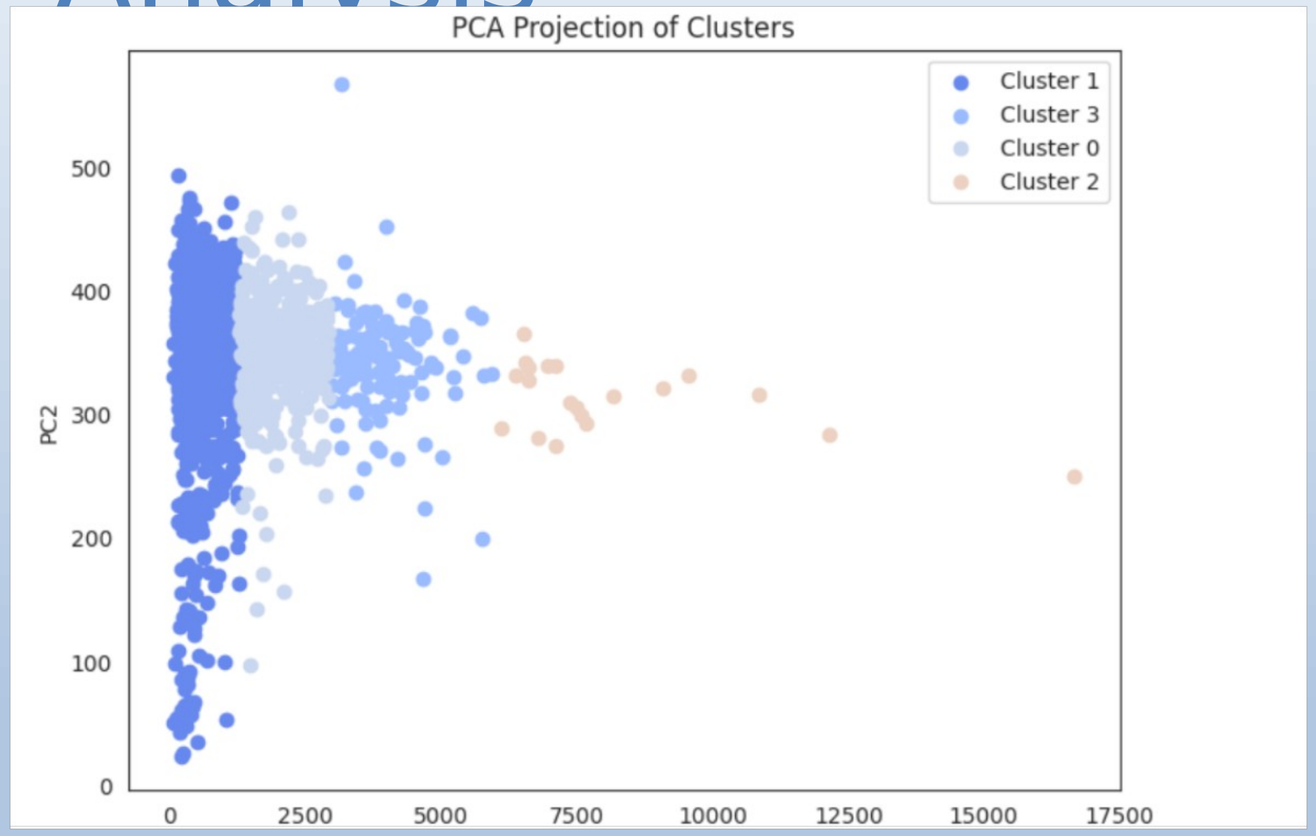
Cluster	Size	Star Ratings	What It Means
Cluster 0	Biggest group	Mostly ⭐4 and ⭐5	Well-rated hotels with strong reviews
Cluster 1	Medium size	⭐4, some ⭐3 and ⭐5	Hotels with mixed or average reviews
Cluster 2	Medium size	⭐4, some ⭐3 and ⭐5	Similar to Cluster 1, but with small differences
Cluster 3	Smallest group	Only ⭐3 and ⭐4	Possibly less popular or more unique hotels

We calculated the Silhouette to see how well hotels fit into their clusters.
Score ranges from -1 to 1
Higher score ->better clusters
Helps us check if k = 4 was a good choice
Our result: Silhouette Score = 0.754
The clusters are clear and well-separated Hotels are grouped well
Our features worked well for clustering hotel reviews

Silhouette Score

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

PCA-Principal Component Analysis



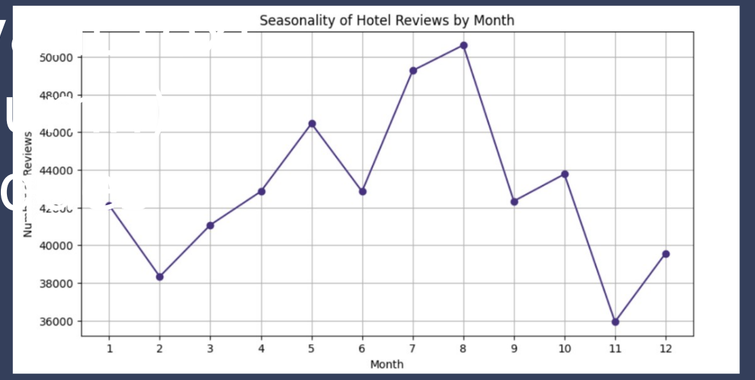
- PCA Projection of Clusters.
- This scatter plot shows how our data points are grouped into 4 clusters.
- Each point represents a hotel or review, and the colors show which cluster it belongs to.
- We used PCA to reduce the dataset to 2 dimensions (PC1 and PC2) so we can visualize it.

- Key Insights:
- Clusters 0, 1, and 3 are close to each other → They likely represent similar types of hotels or reviews.
- Cluster 2 is far away → This group is very different from the others (could be hotels with very high review counts or unique features).

Limitations and further Analysis

Limitations:

- Missing values: Too many blanks and null values making the dataset unreliable and biased
- Unstructured text: Free-form reviews require a lot of pre-processing
- Irrelevant Data: Non-informative content (irrelev
  like positive revies in the 'Negative_Reviews' colu
- Large Volume: Required special tools and technic



Further Analysis:

- Utilize cross-validation for more robust model evaluation
- We could also explore the seasonality of reviews to see if there are patterns across months or seasons.
- For example, do hotels receive more negative reviews in summer due to overcrowding?
- A next step could be to apply our approaches to the full

THANK YOU

Slide 5: Methodology
- **Text Cleaning:** Already lowercase, no punctuation/unicode
- **NLP Techniques:** Tokenization, Word Count, TF-IDF
- **Modeling:** Regression, Clustering, Classification
- **Visualization Tools:** Plotly, Seaborn, WordClouds, Folium

Slide 6: Sample Visualizations
- Map of hotel locations with Average_Score color-coded
- Word clouds for positive vs. negative reviews
- Boxplot of Reviewer_Score by Reviewer_Nationality
- Bar chart of top tags used by reviewers

Slide 4: Possible Analysis Goals
- Sentiment analysis on review texts
- Identify which words are correlated with high/low scores
- Clustering hotels based on guest preferences
- Explore relationship between nationality and score
- Build a simple recommendation system