Analyzing Agoda's Customer Booking Behavior Using Big Data Tools

1 Tool Selection and Methodology

1.1 Apache Spark for Modeling

Apache Spark is an open-source distributed big data processing framework (Cheng et al., 2021). Originally created by Mater Zaharia from the University of California, Berkeley as a faster alternative to MapReduce in 2009, the codebase was later donated to the Apache Software Foundation (Woodie, 2019). According to the Apache Software Foundation, 80% of the Fortune 500 companies use Apache Spark, and there are over 2,000 contributors to the open-source project from industry and academia (The Apache Software Foundation, 2025).

There are several characteristics of Apache Spark that contribute to its widespread popularity:

1. **Efficiency and speed:**

Apache Spark's distributed computing capabilities enable it to scale linearly. This is crucial for Big Data applications such as this project given the extensiveness of historical customer booking data. By leveraging its cluster resources, computations performed during model development can be completed quickly. Furthermore, Spark processes data in memory, which benefits machine learning workflows as they often involve iterative, repetitive tasks, such are hyperparameter tuning (Ratnala et al., 2024).

2. **Ease of Integration with Hadoop Ecosystem**

Spark integrates seamlessly with the Hadoop ecosystem, allowing it to leverage existing Hadoop infrastructure and tools. This compatibility ensures that Spark can handle both batch and streaming data, providing flexibility in how it processes and analyzes data (Salamkar et al., 2021).

3. **Language integration and machine learning support**

Spark supports multiple programming languages, including Python, Scala, Java, and R, making it accessible to a wide range of developers. Additionally, Spark's robust machine learning library, MLlib, provides built-in algorithms and tools for scaling machine learning tasks, such as clustering, regression, and classification, which enhances its suitability for data-driven projects like this one (Rajpurohit et al., 2024).

4. **Real-Time Data Processing:**

Although this project focuses on historical data, Spark's ability to process real-time streaming data is a key strength, as evidenced by its use in time-sensitive applications like heart arrhythmia detection (Salamkar et al., 2021). In the future, this capability can be leveraged to analyze and model real-time booking patterns, providing dynamic insights into customer behavior and enabling more responsive decision-making.

In summary, Apache Spark is an ideal tool for machine learning tasks in big data analytics due to its flexible, fast, and easy-to-use design. These characteristics of Apache Spark form the rationale for using it for model development in this project.

The steps for installing Spark on an Ubuntu machine are outlined below:

a. Download, extract, and move Apache Spark to the /opt/spark directory on Ubuntu by running the following script:

```
1. wget https://downloads.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz.sha512
2. tar xvf spark-*.tgz
3. sudo mv spark-3.5.3-bin-hadoop3 /opt/spark
```

b. Set up the environment variables for Apache on the Ubuntu machine by adding the following to .bashrc before running *source .bashrc*:

```
1. export SPARK_HOME=/opt/spark
2. export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
3. export PYSPARK_PYTHON=/usr/bin/python3
```

c. Run *pyspark* in the terminal to run Apache Spark via the Python shell. Alternatively, .py scripts can be directly submitted to Apache Spark via the terminal by running the following command in the directory containing the .py script:

```
1. spark-submit example_script.py
```

d. Install the required dependencies to perform K-Means clustering and regression:

```
1. sudo apt-get install python3-matplotlib python3-pandas python3-seaborn
```

As described in the overview of tools, two machine learning tasks were undertaken in this project using Spark, namely:

- K-Means clustering: To perform customer segmentation for a better understanding of customer booking behavior and demographics
- Regression models: To build a dynamic pricing model

1.2 K-Means Clustering Methodology

The goal of K-Means clustering in this project is customer segmentation, grouping Agoda customers based on shared booking behaviors. This segmentation enables more targeted marketing campaigns, improved customer service, and the development of tailored products and services.

Customer segmentation can be based on four key dimensions (Kansal et al., 2018):

- Geographic: Region, population density, or climate.
- Demographic: Age and family size,
- Psychographic: Lifestyle variables like interests and attitudes.
- Behavioral: Customer behavior toward products, such as loyalty or purchase readiness.

This project focuses on behavioral segmentation due to the dataset's emphasis on booking-related features. While some demographic and geographic data are present, K-Means clustering is less effective with categorical data because it relies on distance metrics like Euclidean or Manhattan, which are better suited for numerical variables (Dinh et al., 2021; M. Ghazal et al., 2021). Numerical encoding of categorical variables often fails to capture their true relationships, leading to suboptimal clustering. Therefore, this project focuses on analyzing Agoda customers' booking behavior to achieve meaningful segmentation.

After installing Apache Spark on the Ubuntu machine, K-Means clustering was performed following the steps detailed below:

Step 1: Data Preprocessing

Although the data was already cleaned in the initial data preprocessing step, further processing was required to represent the data at a user level and ensure the data was in a suitable format for K-means clustering. The key steps involved are:

- Data aggregation: The flights and hotels datasets were summarized at user-level to obtain *age*, *avg_flight_price*, *total_mileage*, *total_flight_price*, *total_flights*, *avg_flight_distance*, *avg_flight_time*, *total_days_hotel*, *total_hotel_price*, *avg_hotel_price_daily*, and *total_hotels* using SQL on Apache Hive
- Data joins: The aggregated users, flights, and hotels tables were joined using SQL on Apache Hive, and a column was created for *total_price*.
- Encoding categorical features: *gender* and *company* were encoded using *StringIndexer* and *OneHotEncoder* using PySpark
- Feature selection: Irrelevant, intermediate, or redundant columns were dropped after transformation using PySpark.
- Feature engineering: Feature vectors were created using *VectorAssembler* and normalized using *StandardScaler* using PySpark.

Table 1.1 SQL commands for data preprocessing

Data Aggregation for Flights	Data Aggregation for Hotels	Data Joins
`CREATE TABLE agg_flights AS SELECT userCode, ROUND(AVG(price), 2) AS avg_flight_price,`	`CREATE TABLE agg_hotels AS SELECT userCode, SUM(days) AS total_days_hotel, ROUND(SUM(total), 2) AS total_hotel_price,`	`CREATE TABLE agoda_customers AS SELECT a.code, a.gender, a.age, a.company, b.avg_flight_price, b.total_mileage, b.total_flight_price, b.total_flights, b.avg_flight_distance, b.avg_flight_time, c.total_days_hotel,`

```
ROUND(SUM(distance),2)     AS
total_mileage,
       ROUND(SUM(price),2)
AS total_flight_price,
       COUNT(*)            AS
total_flights,
       ROUND(AVG(distance),
2) AS avg_flight_distance,
       ROUND(AVG(time), 2) AS
avg_flight_time
FROM flights
GROUP BY userCode;
```

```
       ROUND(SUM(total)     /
SUM(days),       2)       AS
avg_hotel_price_daily,
       COUNT(*)            AS
total_hotels
FROM hotels
GROUP BY userCode;
```

```
c.total_hotel_price,
c.avg_hotel_price_daily, c.total_hotels,
b.total_flight_price+c.total_hotel_price
AS total_price
FROM users a
JOIN agg_flights b
ON (a.code = b.userCode)
JOIN agg_hotels c
ON (a.code = c.userCode);
```

Table 1.2 Python Script submitted to Apache Spark for data preprocessing

Encoding Categorical Features

```
# Count occurrences of each unique value in gender
dataset.groupBy("gender").count().show()

# Count occurrences of each unique value in company
dataset.groupBy("company").count().show()

# Index the gender column
indexer       =       StringIndexer(inputCol="gender",
outputCol="genderIndex")
dataset = indexer.fit(dataset).transform(dataset)

# One-hot encode the indexed column
encoder     =     OneHotEncoder(inputCol="genderIndex",
outputCol="genderVec")
dataset = encoder.fit(dataset).transform(dataset)

# Drop the original 'gender' column and index column
dataset = dataset.drop("gender", "genderIndex")
```

Data Joins

```
vec_assembler    =    VectorAssembler(inputCols    =
dataset.columns, outputCol='features')

final_data = vec_assembler.transform(dataset)

final_data.printSchema()

scaler = StandardScaler(inputCol="features",
          outputCol="scaledFeatures",
          withStd=True,
          withMean=False)

# Compute summary statistics by fitting the
StandardScaler
scalerModel = scaler.fit(final_data)
```

Step 2: Cluster Initialization and Model Selection

After preprocessing, the Elbow Method and the Silhouette Score were used to determine the optimal number of clusters (k) to be used for K-Means clustering. The Elbow Method obtains the optimal k value by identifying the point where adding more clusters only results in a marginal decrease in the Within-Cluster Sum of Squares (WCSS). WCSS indicates how spread out the data points are in each cluster (Gul & Rehman, 2023). Meanwhile, the Silhouette Score measures the quality of clustering for different k values (Kossakov et al., 2024).

Table 1.3 Python Script submitted to Apache Spark for cluster initialization

Elbow Method

```
# Elbow Method for Optimal K

# Calculate cost and plot
cost = np.zeros(10)

for k in range(2,10):
    kmeans                                          =
KMeans().setK(k).setSeed(1).setFeaturesCol('features')
    model = kmeans.fit(final_data)
    cost[k] = model.summary.trainingCost

# Plot the cost
df_cost = pd.DataFrame(cost[2:])
df_cost.columns = ["cost"]
new_col = [2,3,4,5,6,7,8, 9]
df_cost.insert(0, 'cluster', new_col)

# Saving the values of WCSS
df_cost.to_csv("elbow_method_cost.csv", index=False)
```

Silhouette Score

```
# Silhouette Score

silhouette_scores = []  # List to store silhouette
scores and k values

evaluator                                           =
ClusteringEvaluator(predictionCol='prediction',
featuresCol='scaledFeatures',
metricName='silhouette',
distanceMeasure='squaredEuclidean')

# Loop to calculate silhouette scores for different
k values
for i in range(2, 10):
    kmeans  =  KMeans(featuresCol='scaledFeatures',
k=i, seed=123)
    model = kmeans.fit(final_data)
    predictions = model.transform(final_data)
    score = evaluator.evaluate(predictions)
    silhouette_scores.append({'k':                i,
'silhouette_score': score})
```

```
    print('Silhouette Score for k =', i, 'is', score)

# Create a pandas DataFrame from the silhouette_scores list
silhouette_df = pd.DataFrame(silhouette_scores)
```

Step 3: Analysis of Clusters from K-Means

Upon determining the best k values, the clusters were analyzed by visualizing the clusters via cluster center location plots and pair plots using Python libraries like seaborn and matplotlib, which are supported by Spark. These analyses and visualizations provide an understanding of each customer cluster and what they represent in the context of Agoda's business.

2 Results and Analysis

As stated previously, the aim of K-Means clustering in this project is to create meaningful customer segmentation based on their shared characteristics in support of Agoda's marketing campaigns, customer service, and new product development. The results of K-Means Clustering are detailed below:

2.1 Model Selection

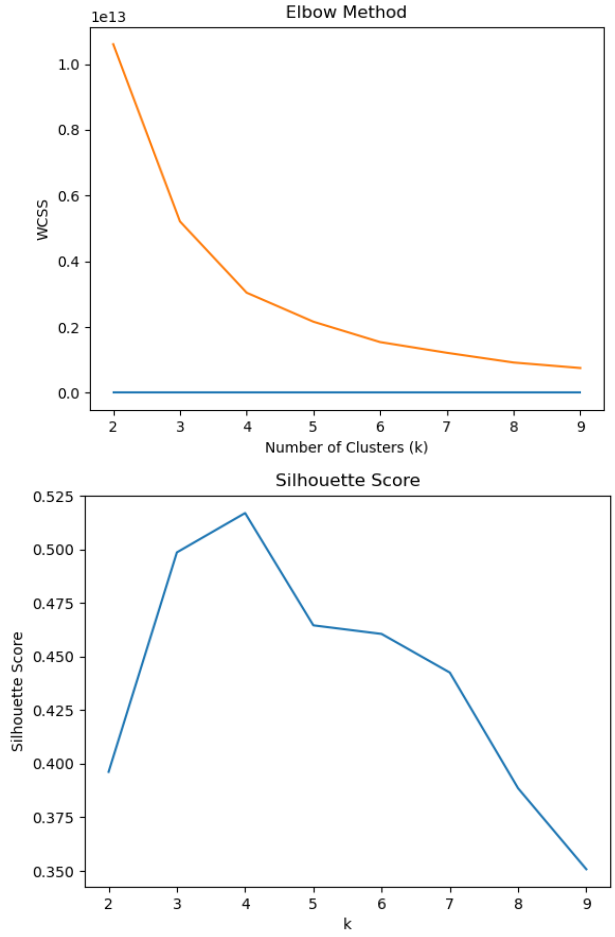

Figure 2.1. (a) Elbow Method plot (b) Silhouette Score plot

From Figure 2.1(a), the Elbow Method locates the elbow at k=4, indicating that the reduction in WCSS beyond k=4 is marginal. On the other hand, the Silhouette Score plot in Figure 2.1(b) shows that the Silhouette Score is the highest at k=4, which means that the cluster quality is the best at a k value of 4. Hence, k=4 was selected as the optimal value for the number of clusters.

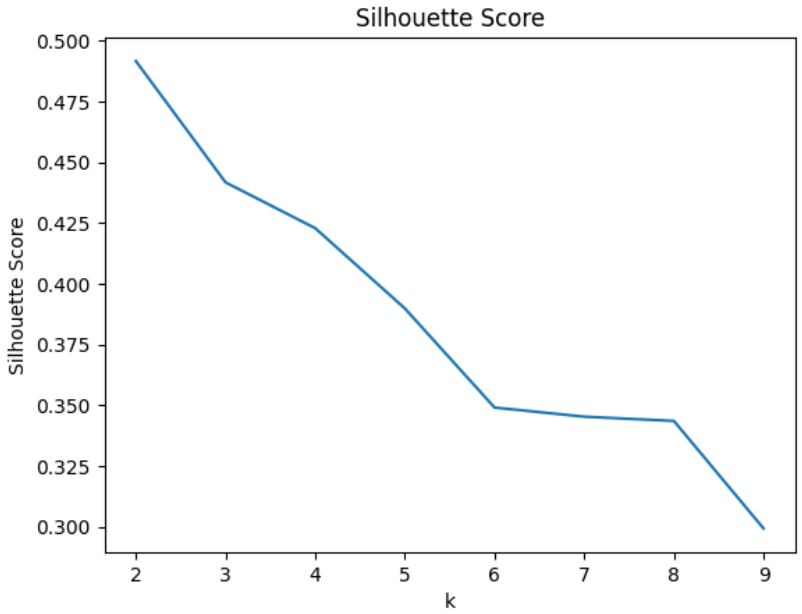

Figure 2.2. Silhouette Score plot upon inclusion of gender as a feature

In the preprocessing step for K-Means clustering, categorical features like gender and company were encoded to numerical values. However, closer examination revealed that the inclusion of these features resulted in poorer clustering performance, as demonstrated by the lower Silhouette Scores in Figure 2.2. This suggests that these categorical features are not contributing meaningful information to clustering or potentially introducing noise. Hence, all categorical columns (including the encoded features derived from them) were dropped from the column. Additionally, *userCode*, which represents the unique identifier for each customer was dropped as well to ensure that the clustering is based on relevant features only.

The features included in K-Means Clustering are listed and described in Table 2.1.

Table 2.1. List of features included in K-Means Clustering

Feature	Description
age	The age of the customer
avg_flight_price	The average price of flight tickets purchased by the customer.
total_mileage	The total flight distance traveled by the customer.
total_flight_price	The total expenses paid by the customer for all flight ticket purchases.
total_flights	The total number of flights taken by the customer.
avg_flight_distance	The average distance traveled by the customer per flight.
avg_flight_time	The average travel time per flight for the customer.
total_days_hotel	The total number of days the customer stayed in all hotels.
total_hotel_price	The total expenses paid by the customer for all hotel bookings.
avg_hotel_price_daily	The average daily cost of hotel bookings paid by the customer.
total_hotels	The total number of hotel bookings made by the customer.
total_price	The total expenses paid by the customer for all flight tickets and hotels.

2.2 Cluster Center Analysis

The coordinates of the centers of the four clusters formed are recorded in Table 2.2, while a visual representation of the coordinates of the cluster centers is illustrated in Figure 2.3. The analysis of these cluster center coordinates provides meaningful insight into the contribution of each feature to the clustering of data points and the characteristics of each cluster.

To analyze the cluster centers effectively, the percentage differences of the cluster centers from the average across all clusters were calculated for each feature, as recorded in

Table 2.3. These percentages indicate how much the cluster centers differ from the baseline average for each feature. A positive percentage (highlighted in green) indicates that the cluster center has a higher value for that feature compared to the baseline, while a negative percentage (highlighted in red) indicates a lower value compared to the baseline. The intensity of the color represents the magnitude of the percentage difference.

The percentage difference values of the cluster centers yielded several key observations for the following features:

- total_mileage, total_price, total_flights, total_days_hotel, total_hotel_price, total_hotels:
    - Higher percentages for clusters 0 and 2: Customers in these clusters have traveled more frequently, covered longer total flight distances, and made more hotel bookings, resulting in higher flight and hotel expenses.
    - Lower percentages for clusters 1 and 3: Customers in these clusters have traveled less frequently, covered shorter total flight distances, and made fewer hotel bookings, leading to lower flight and hotel expenses.
- avg_flight_price, avg_flight_price, avg_flight_time:
    - Higher percentages for clusters 2 and 3: Customers in these clusters purchase more expensive flights and typically travel longer distances.
    - Lower percentages for clusters 0 and 1: Customers in these clusters typically purchase cheaper flights and travel shorter distances.
- avg_hotel_price_daily:
    - Slightly higher percentages for clusters 0 and 1: Customers in these clusters may spend slightly more per night during hotel stays.
    - Slightly lower percentages for clusters 2 and 3: Customers in these clusters may spend slightly less per night on hotel stays.
- age: The cluster centers do not appear to show significant differences from each other for this feature.

Based on the observations, the four clusters or segments of customers can be characterized as such:

- Cluster 0: Frequent travellers who typically travel shorter distances.
- Cluster 1: Infrequent travellers who typically travel shorter distances.
- Cluster 2: Frequent travellers who typically travel longer distances.
- Cluster 3: Infrequent travellers who typically travel longer distances.

Table 2.2. Coordinates of the cluster centers

cluster	age	avg_ flight_ price	total_ mileag e	total_ flight_ price	total_ flights	avg_ flight_ distanc e	avg_ flight_ time	total_ days_ hotel	total_ hotel_ price	avg_ hotel_ price_ daily	total_ hotels	total_ price
0	3.37	10.39	2.25	2.52	2.70	5.03	5.03	2.58	2.62	8.58	2.60	2.54
1	3.33	10.27	0.81	0.91	0.99	4.94	4.95	0.93	0.93	8.51	0.93	0.92
2	3.25	12.05	3.13	2.94	2.72	6.97	6.97	2.63	2.48	7.97	2.66	2.92
3	3.28	12.05	1.06	1.00	0.93	6.94	6.94	0.85	0.80	7.86	0.86	0.99

Table 2.3. Percentage difference of each cluster center from the average baseline across clusters for all features

cluster	age	avg_ flight_ price	total_ mileage	total_ flight_ price	total_ flights	avg_ flight_ distance	avg_ flight_ time	total_ days_ hotel	total_ hotel_ price	avg_ hotel_ price_ daily	total_ hotels	total_ price
0	1.9%	-7.1%	23.9%	36.8%	47.4%	-15.8%	-15.8%	47.6%	53.2%	4.3%	47.6%	38.0%
1	0.6%	-8.3%	-55.3%	-50.4%	-46.0%	-17.2%	-17.2%	-47.1%	-45.7%	3.4%	-47.2%	-50.1%
2	-1.8%	7.7%	72.7%	59.4%	48.1%	16.8%	16.7%	50.7%	45.4%	-3.2%	50.7%	58.4%
3	-0.7%	7.7%	-41.3%	-45.8%	-49.4%	16.2%	16.2%	-51.2%	-53.0%	-4.5%	-51.1%	-46.3%

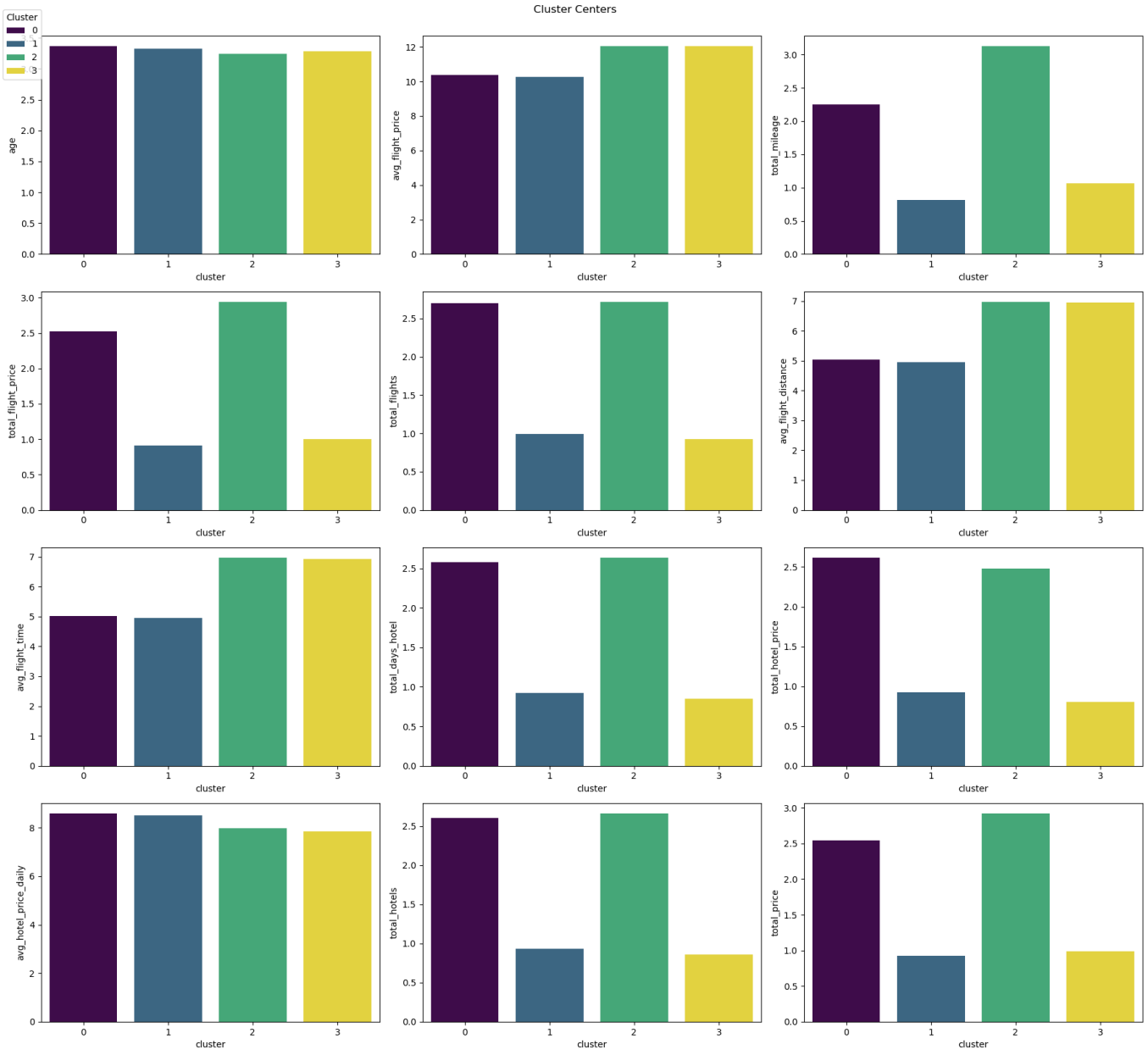

Figure 2.3. Bar plot of cluster center value for each feature.

2.3 2-Dimensional (2D) and 3-Dimensional (3D) Scatter plots

The features selected for 2D and 3D scatter plots in Figure 2.4 are representative of the key characteristics and observations identified via the cluster center analysis. The 2D scatter plot illustrates the strong discriminatory power avg_flight_distance holds while distinguishing between the clusters. This is evident as clusters with higher or lower average flight distances exhibit clear separations along this axis. While total_price does not visually separate some clusters in the scatter plots, its contribution to clustering is evident in the center analysis. It provides context when combined with other features like avg_flight_distance and total_mileage, providing a holistic view of a customer's travel and spending patterns. Meanwhile, the 3D scatter plot further enriches the analysis by integrating total_price and avg_hotel_price_daily into the visualization.

2.4 Pair Plots

The pair plots in Figure 2.5 visualize pairwise relationships between features and clusters, allowing for a visual assessment of clustering quality. In line with the previous cluster center analysis, features like age and avg_hotel_price_daily show limited discriminatory power. The overlapping data points and superimposed distribution curves for these features suggest they do not meaningfully contribute to distinguishing the clusters. While clusters remain visible, significant overlaps make it challenging to characterize them based solely on these features.

- Cluster 0: Frequent travellers who typically travel shorter distances.
- Cluster 1: Infrequent travellers who typically travel shorter distances.
- Cluster 2: Frequent travellers who typically travel longer distances.
- Cluster 3: Infrequent travellers who typically travel longer distances.

2.5 Recommendations

Based on the interpretation of the results from K-Means clustering, Agoda can elevate its business by implementing the following:

1. Tailored Marketing Campaigns

Different marketing campaigns can be created for different clusters of customers.

- Cluster 0 (Frequent travellers who travel shorter distances): Agoda can target these customers with promotions for frequent bookings, such as loyalty programs or discounts for shorter trips . Personalized marketing can focus on offering attractive deals for nearby destinations or weekend getaways (Kisliakova, 2022).
- Cluster 1 (Less frequent travellers, shorter distances): This group may be more sensitive to price. Agoda can target them with promotions on budget-friendly hotels and short-distance flights. Highlighting affordable options, last-minute deals, and local experiences could be more appealing to these customers (Tran, 2024).
- Cluster 2 (Frequent travellers who travel longer distances): Agoda could offer these customers premium deals on international flights or longer-term hotel bookings. Since they tend to travel more frequently and cover long distances, they may appreciate tailored offers for frequent flyer programs, long-haul flight packages, or luxury hotels (Kisliakova, 2022; Tran, 2024).
- Cluster 3 (Less frequent travellers, longer distances): For these customers, Agoda could introduce tailored offers like seasonal promotions for vacations to faraway destinations. Long-term stay offers or package deals that include both flights and hotels might be more effective.

2. Customer Personalization

Agoda can personalize the user experience based on the clusters of frequent and infrequent travellers to improve their experience:

- Frequent travellers: Agoda can provide personalized recommendations for this segment based on their preferences, such as faster check-in, curated lists of hotels suited for frequent stays, or exclusive services (Tran, 2024).
- Infrequent travellers: For customers who travel less frequently, Agoda can provide helpful travel guides, discounts for first-time bookings, or special travel packages for special occasions (e.g., vacations, anniversaries).

3. Pricing Strategy Optimization

Agoda can leverage its understanding of customer spending behaviour to optimize its pricing strategy based on different clusters:

- Cluster 2 and 3 (Higher expenses on average): For these customers who tend to spend more per flight, Agoda can offer upselling opportunities such as premium accommodations, flights with additional services, or exclusive add-ons (e.g., lounge access, airport transfers) (Tran, 2024).
- Cluster 0 and 1 (Lower expenses on average): Agoda can introduce budget-friendly options and targeted promotions that align with the customer's preferences for lower-cost hotels and flights. Providing them with cost-effective packages or offering more affordable options could increase engagement (Tran, 2024).

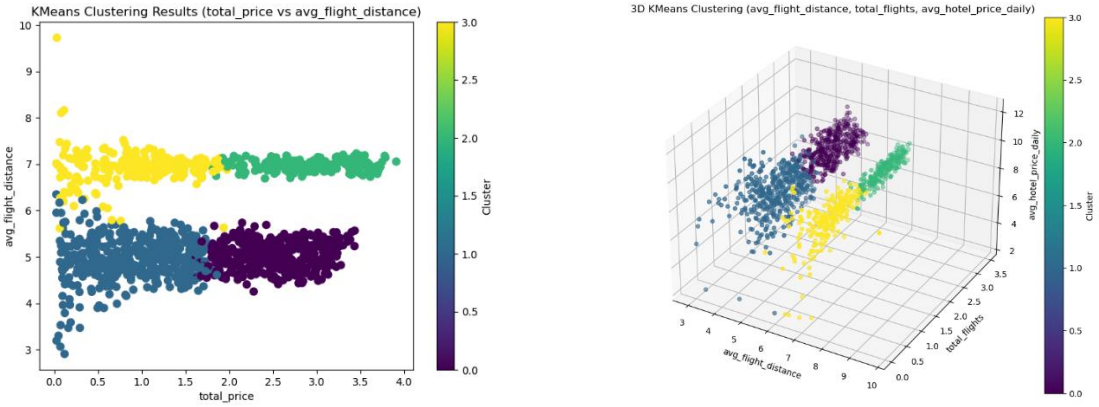

Figure 2.4. 2D and 3D scatter plots of the cluster based on avg_flight_distance, total_price, and avg_hotel_price_daily

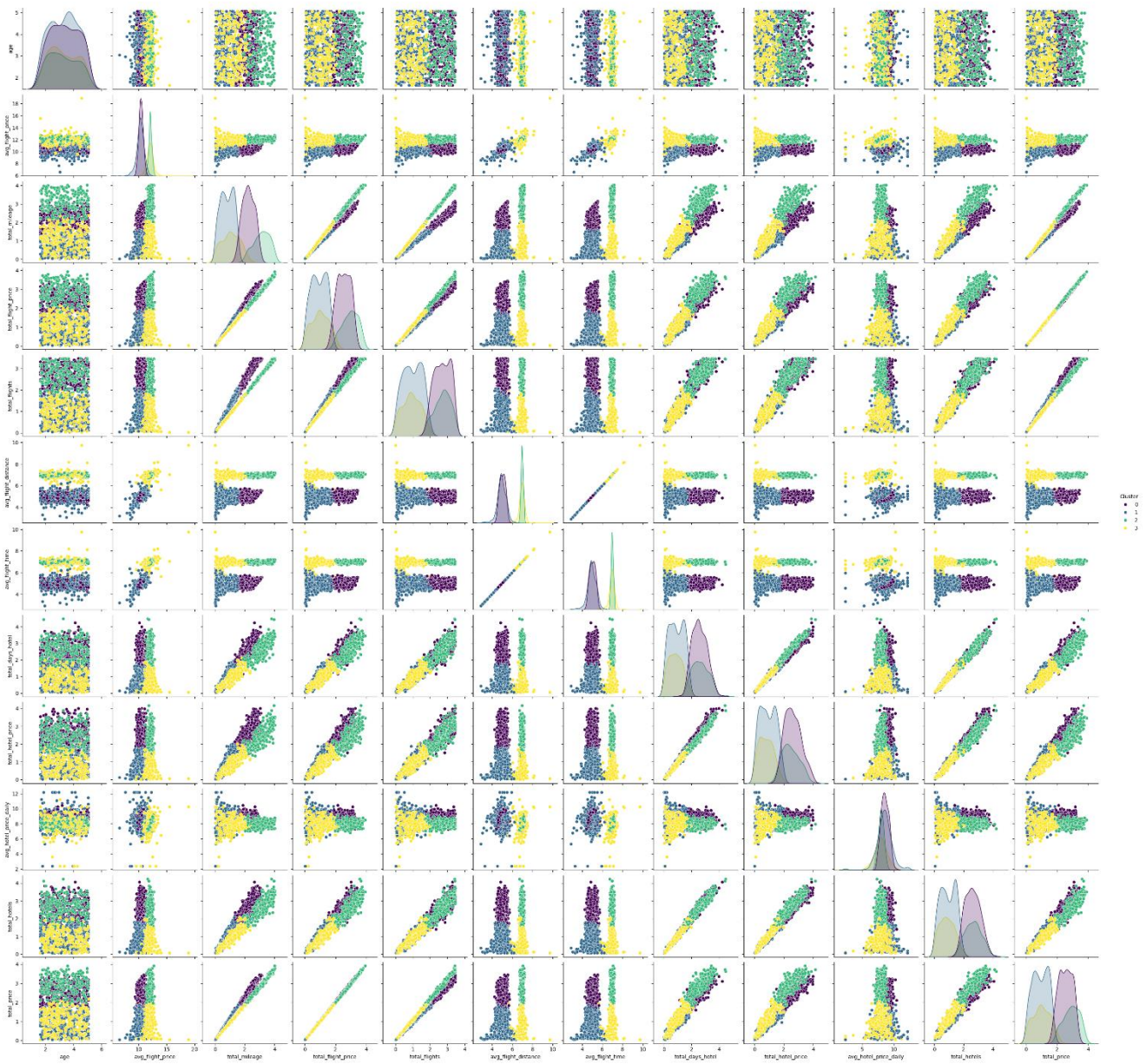

Figure 2.5. Cluster predictions for all pairs of features

3 REFERENCES

1. Cheng, G., Ying, S., Wang, B., & Li, Y. (2021). Efficient Performance Prediction for Apache Spark. *Journal of Parallel and Distributed Computing*, *149*, 40–51. https://doi.org/10.1016/j.jpdc.2020.10.010
2. Dinh, D.-T., Huynh, V.-N., & Sriboonchitta, S. (2021). Clustering mixed numerical and categorical data with missing values. *Information Sciences*, *571*, 418–442. https://doi.org/10.1016/j.ins.2021.04.076
3. Gul, M., & Rehman, M. A. (2023). Big data: An optimized approach for cluster initialization. *Journal of Big Data*, *10*(1), 120. https://doi.org/10.1186/s40537-023-00798-1
4. Kansal, T., Bahuguna, S., Singh, V., & Choudhury, T. (2018). Customer Segmentation using K-means Clustering. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 135–139. https://doi.org/10.1109/CTEMS.2018.8769171
5. Kisliakova, I. (2022). *Strategy to increase domestic and international direct bookings for a hotel* [Karelia University of Applied Sciences]. https://www.theseus.fi/bitstream/handle/10024/780989/Kisliakova_Iuliia.pdf?sequence=5
6. Kossakov, M., Mukasheva, A., Balbayev, G., Seidazimov, S., Mukammejanova, D., & Sydybayeva, M. (2024). Quantitative Comparison of Machine Learning Clustering Methods for Tuberculosis Data Analysis. *CIEES 2023*, 20. https://doi.org/10.3390/engproc2024060020
7. M. Ghazal, T., Zahid Hussain, M., A. Said, R., Nadeem, A., Kamrul Hasan, M., Ahmad, M., Adnan Khan, M., & Tahir Naseem, M. (2021). Performances of K-Means Clustering Algorithm with Different Distance Metrics. *Intelligent Automation & Soft Computing*, *29*(3), 735–742. https://doi.org/10.32604/iasc.2021.019067
8. Rajpurohit, A. M., Kumar, P., Kumar, R. R., & Kumar, R. (2024). A Review on Apache Spark. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4492445
9. Ratnala, A. K., Inampudi, R. K., & Pichaimani, T. (2024). Evaluating Time Complexity in Distributed Big Data Systems: A Case Study on the Performance of Hadoop and Apache Spark in Large-Scale Data Processing. *Journal of Artificial Intelligence Research and Applications*, *4*(1), Article 1.
10. Salamkar, M. A., Allam, K., & Immaneni, J. (2021). The Big Data Ecosystem: An overview of critical technologies like Hadoop, Spark, and their roles in data processing landscapes. *Journal of AI-Assisted Scientific Discovery*, *1*(2), Article 2.
11. The Apache Software Foundation. (2025). *Apache Spark™—Unified Engine for large-scale data analytics*. https://spark.apache.org/
12. Tran, X. (2024). *Strategies of Revenue Management*. https://pressbooks.uwf.edu/revenuemanagementillustrated/chapter/chapter-5/
13. Woodie, A. (2019, March 8). *A Decade Later, Apache Spark Still Going Strong*. BigDATAwire. https://www.bigdatawire.com/2019/03/08/a-decade-later-apache-spark-still-going-strong/