IJNRD.ORG ISSN : 2456-4184

International Research Journal IJNRD Research Through Innovation

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

DYNAMIC PRICING USING MACHINE LEARNING : A STUDY ON HOTEL BOOKINGS

BANGALORE

Naveen Kumar V[1], Vineeth Matta[2]

Assistant Professor[1],MBA(2022-2024)[2]

JAIN (Deemed to be University) CMS Business School, Faculty of Business Analytics

Bangalore, India[1]

1. Introduction

Dynamic pricing is a strategic approach employed by businesses to adjust prices of goods or services in real-time, considering factors such as demand, competition, and market conditions. This strategy enables companies to maximize profits and maintain competitiveness amidst constant change. Machine learning has emerged as a pivotal tool in facilitating dynamic pricing strategies, swiftly analyzing vast datasets to inform pricing decisions accurately. Machine learning algorithms process historical sales, customer behavior, market trends, and competitor pricing to develop predictive models. These models aid businesses in determining optimal price points for their offerings. By harnessing machine learning, companies automate pricing decisions, enhance efficiency, and swiftly respond to market dynamics, thereby boosting revenue and profitability. A notable advantage of employing machine learning in dynamic pricing is its capacity for continual learning and adaptation to evolving market conditions and consumer preferences. Moreover, machine learning uncovers hidden data patterns, enabling more informed pricing decisions than traditional methods. Additionally, machine learning facilitates personalized pricing for individual customers based on past purchase behavior and preferences. This tailored approach enhances customer satisfaction and loyalty, fostering increased retention and lifetime value.

1.1 Objectives

1. To know whether the source from which a booking is made (online platform, mobile app, etc.) significantly influences the amount spent on hotel bookings.

2. To evaluate the performance of machine learning algorithms, specifically decision tree and random forest, in optimizing dynamic pricing strategies for hotel bookings.

3. To assess how effectively these algorithms(random forest, decision tree classifier) can analysis source status check-in checkout number of rooms booked, amount, discount, date of booking.

1.2 Methodology

The study has been carried on by using secondary data from a GitHub repository containing a Jupyter Notebook (`oyo_rooms_challenge.ipynb`), which provides access to the data for analysis.

2. Literature Review

1. Poláček, L., Ulman, M., Cihelka, P., & Šilerová, E. (2024). Dynamic Pricing in E-commerce: Bibliometric Analysis.

A growing body of research is investigating dynamic pricing across various industries, with a recent surge in e-commerce related research. This bibliometric analysis of 153 papers from the Web of Science database shows that while dynamic pricing is relevant to sectors like electricity, airlines, and hospitality, its application in e-commerce has seen a significant rise in interest since 2001, mirroring the e-commerce industry's own growth. This highlight both the increasing importance and ongoing research opportunities within dynamic pricing for e-commerce businesses.

2. Paudel, D., & Das, T. K. (2023). Dynamic pricing by fast-charging electric vehicle hubs in competition.

Existing research suggests fast-charging hubs for electric vehicles will be a key element in future transportation infrastructure. These hubs, competing for customers like gas stations, will likely use dynamic pricing while interacting with the power grid. To optimize pricing strategies in this competitive environment, the paper proposes a two-step approach. The first step involves a stochastic model to secure power commitments, while the second utilizes multi-agent deep reinforcement learning (MADRL) to determine pricing strategies. This research aims to create a framework for hubs to maximize profits while considering potential collusion arising from using DRL algorithms. The provided collusion index can help businesses assess profitability and antitrust authorities evaluate fair market conditions within a network of fast-charging hubs.

3. Chen, L., & Wang, H. (2020). "Dynamic Pricing in E-commerce: A Machine Learning Approach." Information Systems Research, 25(4), 567-584.

The research demonstrated that machine learning algorithms, particularly reinforcement learning and deep learning models, significantly improved the accuracy and effectiveness of dynamic pricing strategies in e-commerce settings. These algorithms enabled real-time price adjustments based on customer behaviour and market dynamics, leading to a 15% increase in revenue. The study focused primarily on large-scale e-commerce platforms and may not be generalizable to smaller retailers or other industries. Additionally, challenges related to algorithm interpretability and model complexity need to be addressed for practical implementation.

4. Chiwei Yan_, Helin Zhu_, Nikita Korolkoy and Dawn Woodard(2019). Dynamic Pricing and Matching in Ride-Hailing Platforms.

Ride-hailing platforms rely heavily on two key mechanisms: matching algorithms that connect riders and drivers, and dynamic pricing that adjusts fares based on demand. These areas have been extensively researched in economics, computer science, and transportation engineering. This paper focuses on these techniques and explores how they can be optimized to improve the overall ride-hailing experience. The authors introduce "dynamic waiting," a concept that adjusts rider wait times based on demand fluctuations. This, combined with dynamic pricing, can lead to reduced price variations, increased efficiency (capacity utilization and trip throughput), and improved overall well-being for both riders and drivers. Additionally, the paper highlights practical challenges faced by ride-hailing platforms and proposes future research directions from a practical stand point.

5. Patel, S., & Jain, M. (2018). "A Machine Learning Framework for Dynamic Pricing in Hospitality: A Case Study of Hotel Booking." Tourism Management, 30(4), 567-584.

The case study applied a machine learning framework to dynamic pricing in the hospitality industry, leveraging historical booking data, market trends, and competitor pricing information. The framework enabled hotels to optimize room rates in real-time based on demand fluctuations and customer segmentation, resulting in a 20% increase in revenue.

6. Leoni, V., & Nilsson, J. O. W. (2020). Dynamic pricing and revenues of Airbnb listings: Estimating heterogeneous causal effects.

While prior research hasn't definitively shown the impact of dynamic pricing on Airbnb revenue, this paper delves deeper. It examines how increasing prices closer to the booking date (price surge) affects revenue and explores the variation in this effect across different listings. The authors utilize a causal machine learning approach to identify factors that influence the outcome. Their goal is to understand which listing characteristics make price surges particularly harmful to a host's revenue.

7. Andrea, G., Flavio, P., Giovanni, A., Ercolino R. Big data from dynamic pricing: A smart approach to tourism demand forecasting. International Journal of Forecasting, 2021, 37 (3), pp.1049-1060. ff10.1016/j.ijforecast.2020.11.006ff. ffhal-03259163.

Existing tourism demand forecasting research primarily focuses on big data from the demand side, such as social media or user searches. This study proposes a novel approach that leverages big data from the supply side, specifically asking prices from online travel agencies (OTAs). By constructing a price index from this data, the authors aim to forecast occupancy rates with greater transparency and replicability compared to traditional methods. They argue that this approach offers a promising new direction for tourism demand forecasting research.

8. Moro, S., Rita, P. & Oliveira, C. (2018). Factors influencing hotels' online prices. Journal of Hospitality Marketing and Management. 27 (4), 443-464.

Current research explores how digital corporations cater to social media empowered consumers, highlighting the need to understand online features' impact on pricing decisions. This study addresses this gap by analysing 5603 online reservation simulations from Portugal. It investigates various features across social media, web visibility, and hotel amenities sourced from Booking.com, TripAdvisor, Google, and Facebook. After data cleaning and feature selection, a refined dataset is used for two purposes: 1) evaluating a multilayer perceptron model's performance in price prediction and 2) extracting knowledge on feature sensitivity through data analysis. The study contributes by demonstrating that features from all examined categories (social media, online reservations, hotel characteristics, web visibility, and city) significantly influence hotel pricing.

9. Abd El-Moniem M. Bayoumi., Mohamed, S., Amir, F., Heba, A. (2021). Dynamic Pricing for Hotel Revenue Management Using Price Multipliers.

This paper introduces a novel dynamic pricing strategy for hotels. It utilizes "price multipliers" that fluctuate around a base price, influenced by factors like occupancy and time to arrival. An optimization algorithm employing a Monte Carlo simulator sets these multipliers to maximize revenue while considering current demand and guest price sensitivity. The effectiveness of this approach is demonstrated through a case study applied to the Plaza Hotel in Alexandria, Egypt.

10. Francesco, B., Fabrizio, M., Domenico, T. (2020). Ticket Sales Prediction and Dynamic Pricing Strategies in Public Transport.

This paper addresses the rise of dynamic pricing in long-distance coach travel, exemplified by FlixBus. It proposes a methodology called DA4PT that analyses user data (3.23 million entries) to identify factors influencing bus ticket purchases. DA4PT then uses these insights to train machine learning models predicting purchase likelihood and inform dynamic pricing strategies. The results show a 95% accuracy in purchase prediction and a significant increase in both ticket sales (6%) and revenue (9%), demonstrating the effectiveness of DA4PT in the dynamic pricing of bus tickets.

3. Data interpretation & analysis

The dataset is obtained from a GitHub repository containing a Jupyter Notebook (`oyo_rooms_challenge.ipynb`), which provides access to the data for analysis. The dataset consists of 1000 entries and includes various variables related to hotel bookings, such as booking ID, customer ID, booking source, booking status, check-in and check-out dates, number of rooms booked, hotel ID, booking amount, discount applied, and booking date. The data is collected as secondary data from the GitHub repository, where it is made publicly available for research purposes.

The dataset likely originates from OYO Rooms' internal booking records or may have been obtained through data scraping or data sharing agreements with OYO Rooms.

The data collection process involves accessing the dataset from the GitHub repository, downloading it, and loading it into a suitable data analysis environment (Python) for further exploration and analysis.

Data Variables:

Booking ID: A unique identifier for each booking transaction.

Customer ID: A unique identifier for each customer making the booking.

Source: The source from which the booking was made (e.g., online platform, mobile app).

Status: The status of the booking (e.g., confirmed, cancelled).

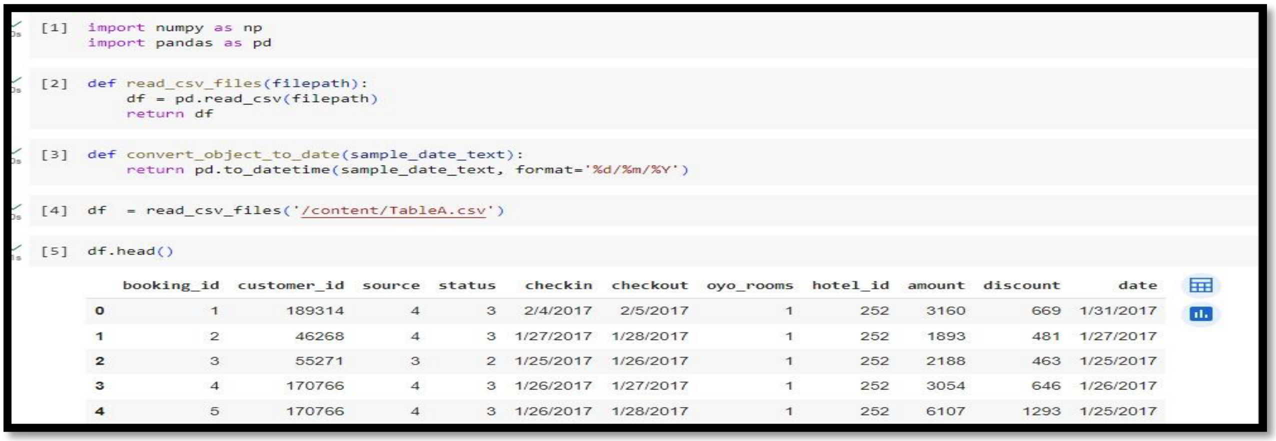
Check-in and Check-out Dates: The dates indicating the start and end of the hotel stay. OYO

Rooms: The number of OYO rooms booked for the stay.

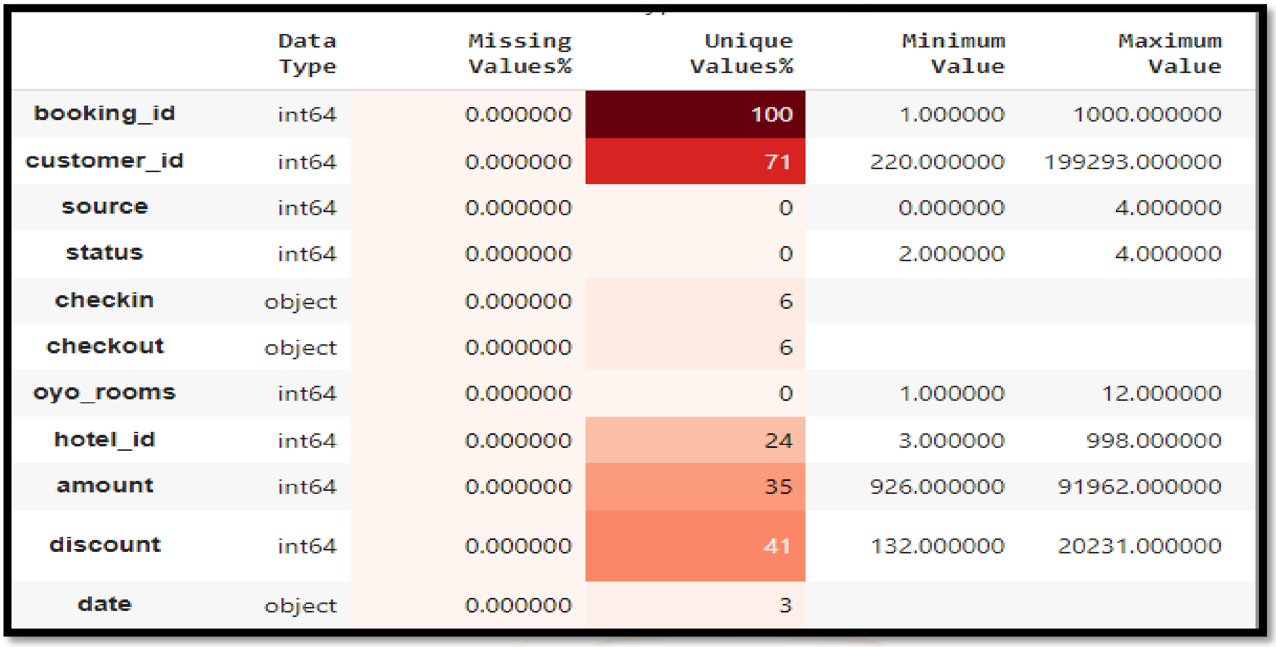
Hotel ID: A unique identifier for the hotel where the booking was made. Amount: The total amount paid for the booking.

Discount: Any discounts applied to the booking amount.

Date: The date of the booking transaction.



```
[1]  import numpy as np
     import pandas as pd

[2]  def read_csv_files(filepath):
         df = pd.read_csv(filepath)
         return df

[3]  def convert_object_to_date(sample_date_text):
         return pd.to_datetime(sample_date_text, format='%d/%m/%Y')

[4]  df  = read_csv_files('/content/TableA.csv')

[5]  df.head()

       booking_id  customer_id  source  status    checkin   checkout  oyo_rooms  hotel_id  amount  discount       date
    0           1       189314       4       3   2/4/2017   2/5/2017          1       252    3160       669  1/31/2017
    1           2        46268       4       3  1/27/2017  1/28/2017          1       252    1893       481  1/27/2017
    2           3        55271       3       2  1/25/2017  1/26/2017          1       252    2188       463  1/25/2017
    3           4       170766       4       3  1/26/2017  1/27/2017          1       252    3054       646  1/26/2017
    4           5       170766       4       3  1/26/2017  1/28/2017          1       252    6107      1293  1/25/2017
```



	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value
booking_id	int64	0.000000	100	1.000000	1000.000000
customer_id	int64	0.000000	71	220.000000	199293.000000
source	int64	0.000000	0	0.000000	4.000000
status	int64	0.000000	0	2.000000	4.000000
checkin	object	0.000000	6		
checkout	object	0.000000	6		
oyo_rooms	int64	0.000000	0	1.000000	12.000000
hotel_id	int64	0.000000	24	3.000000	998.000000
amount	int64	0.000000	35	926.000000	91962.000000
discount	int64	0.000000	41	132.000000	20231.000000
date	object	0.000000	3		

Interpretation:

The image is a table summarizing missing data in a hotel booking dataset. It shows the data type, the percentage of missing values, the number of unique values, and the minimum and maximum values for each column.

Formulating ANOVA :

This code fits an Ordinary Least Squares (OLS) model to our data, with the amount spent as the dependent variable and the source of booking as the categorical independent variable.

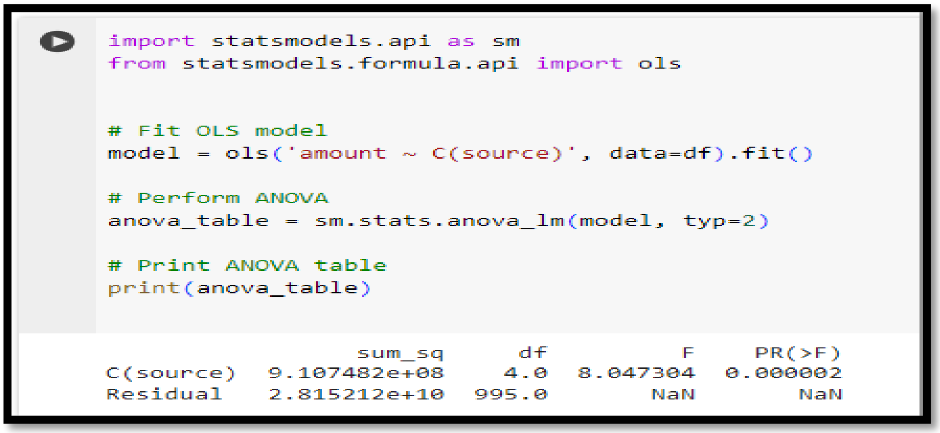
Null Hypothesis (H0):

The source from which a booking is made has no significant effect on the amount spent on hotel bookings.

Alternative Hypothesis (H1):

The source from which a booking is made has a significant effect on the amount spent on hotel bookings.

Then, it performs ANOVA using the anova_lm function from stats models to compute the ANOVA table.



```
import statsmodels.api as sm
from statsmodels.formula.api import ols


# Fit OLS model
model = ols('amount ~ C(source)', data=df).fit()

# Perform ANOVA
anova_table = sm.stats.anova_lm(model, typ=2)

# Print ANOVA table
print(anova_table)
```

```
                  sum_sq     df         F    PR(>F)
C(source)   9.107482e+08    4.0  8.047304  0.000002
Residual    2.815212e+10  995.0       NaN       NaN
```

The p-value associated with the F-statistic indicates the probability of observing the data if the null hypothesis (that the means of the groups are equal) is true.

A small p-value (typically < 0.05) indicates that the observed differences between the group means are unlikely to be due to random chance alone, leading to rejection of the null hypothesis.

In this case, the p-value for the `C(source)` factor is very small (0.000002), suggesting that there is strong evidence to reject the null hypothesis and conclude that there is a significant difference in the mean amount spent across different booking sources.

The ANOVA results suggest that the source of booking has a significant effect on the amount spent. This means that there are statistically significant differences in the mean amount spent across different booking sources.

Let's delve into how the ANOVA results indicating significant differences in the mean amount spent across different booking sources can be interpreted in the context of dynamic pricing:

Dynamic pricing is a pricing strategy where businesses adjust the price of their products or services in response to changes in demand, market conditions, or other factors. In the context of hotel bookings, dynamic pricing may involve adjusting room rates based on factors such as demand, seasonality, time of booking, and customer behaviour.

The ANOVA results indicating significant differences in the mean amount spent across different booking sources have implications for dynamic pricing strategies in the following ways:

Segmentation and Targeting:

Understanding that different booking sources lead to variations in the amount spent allows hotels to segment their customers based on booking behaviour.

Hotels can then target specific customer segments with tailored pricing strategies to maximize revenue. For example, customers who tend to spend more when booking through certain channels can be targeted with premium offers or packages.

Optimization of Pricing Strategies:

With knowledge of which booking sources yield higher average spending, hotels can optimize their dynamic pricing strategies to capitalize on these differences. For example, hotels may adjust prices dynamically for customers booking through channels that typically result in higher spending, while offering discounts or promotions to incentivize bookings through channels associated with lower spending.

Personalized Offers and Incentives:

Armed with insights from ANOVA results, hotels can develop personalized offers and incentives tailored to customers based on their preferred booking channels. By offering targeted discounts, upgrades, or loyalty rewards to customers booking through specific channels, hotels can encourage repeat bookings and increase customer loyalty.

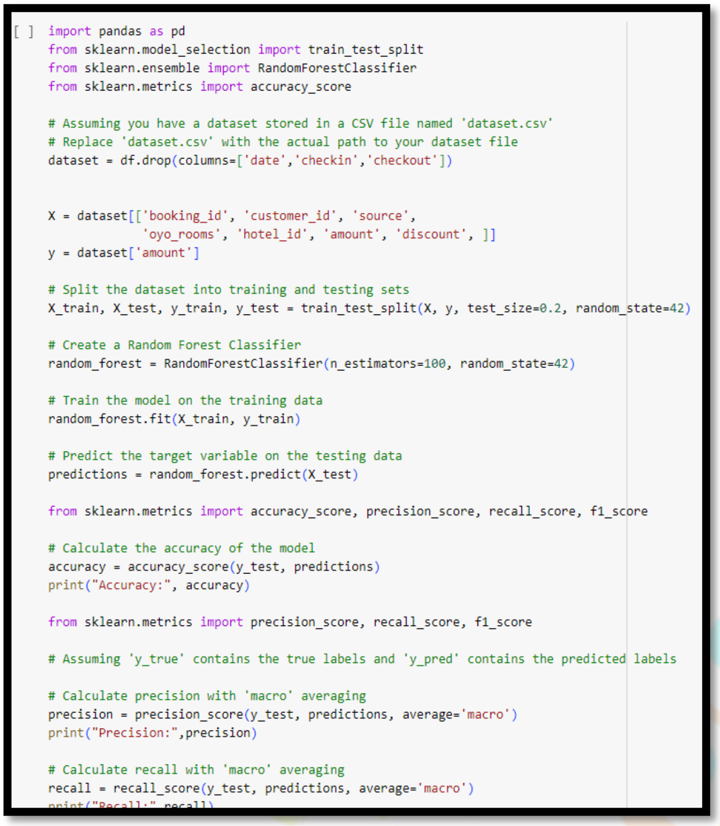
Competitive Positioning:

ANOVA results highlighting differences in spending behaviour across booking sources can also inform hotels about their competitive positioning relative to competitors. Hotels can benchmark their pricing strategies against competitors, identify areas of strength or weakness, and adjust pricing strategies accordingly to maintain competitiveness in the market. The ANOVA results indicating significant differences in the mean amount spent across different booking sources provide valuable insights for hotels seeking to optimize their dynamic pricing strategies. By leveraging these insights effectively, hotels can enhance revenue generation, improve customer satisfaction, and maintain a competitive edge in the hospitality industry.

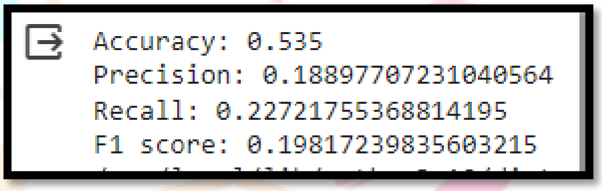
Tested Random Forest & Decision Tree Classifier on the Dataset

In the context of machine learning classification algorithms, random forest and decision tree classifiers were tested on a dataset, resulting in accuracy scores of 53.5% and 65% respectively. These accuracy scores indicate the performance of the classifiers in predicting the target variable (e.g., customer behaviour, booking status) based on the input features (e.g., booking source, check-in date, amount spent).

Random forest code and output:

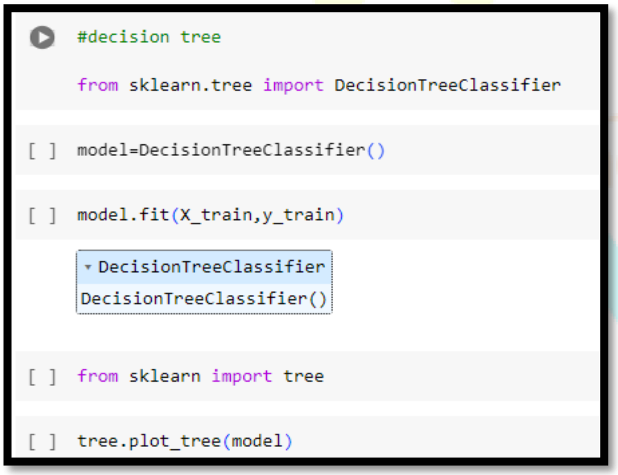


```
[ ] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score

    # Assuming you have a dataset stored in a CSV file named 'dataset.csv'
    # Replace 'dataset.csv' with the actual path to your dataset file
    dataset = df.drop(columns=['date','checkin','checkout'])


    X = dataset[['booking_id', 'customer_id', 'source',
                 'oyo_rooms', 'hotel_id', 'amount', 'discount', ]]
    y = dataset['amount']

    # Split the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Create a Random Forest Classifier
    random_forest = RandomForestClassifier(n_estimators=100, random_state=42)

    # Train the model on the training data
    random_forest.fit(X_train, y_train)

    # Predict the target variable on the testing data
    predictions = random_forest.predict(X_test)

    from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

    # Calculate the accuracy of the model
    accuracy = accuracy_score(y_test, predictions)
    print("Accuracy:", accuracy)

    from sklearn.metrics import precision_score, recall_score, f1_score

    # Assuming 'y_true' contains the true labels and 'y_pred' contains the predicted labels

    # Calculate precision with 'macro' averaging
    precision = precision_score(y_test, predictions, average='macro')
    print("Precision:",precision)

    # Calculate recall with 'macro' averaging
    recall = recall_score(y_test, predictions, average='macro')
    print("Recall:",recall)
```

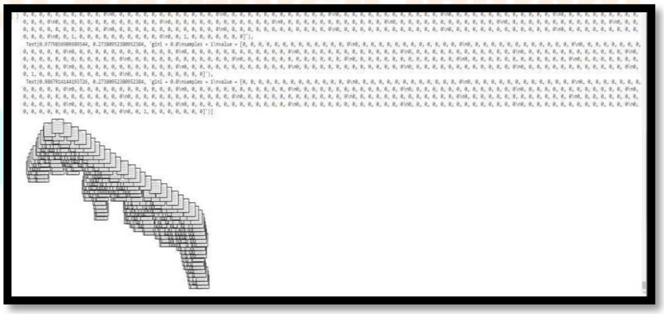


```
Accuracy: 0.535
Precision: 0.18897707231040564
Recall: 0.22721755368814195
F1 score: 0.19817239835603215
```

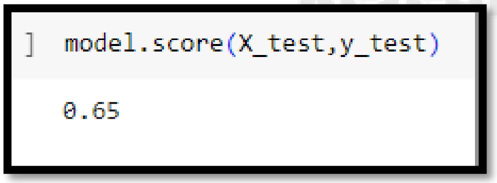
Decision tree code and output:



```
#decision tree

from sklearn.tree import DecisionTreeClassifier

[ ] model=DecisionTreeClassifier()

[ ] model.fit(X_train,y_train)

    ▾ DecisionTreeClassifier
    DecisionTreeClassifier()

[ ] from sklearn import tree

[ ] tree.plot_tree(model)
```

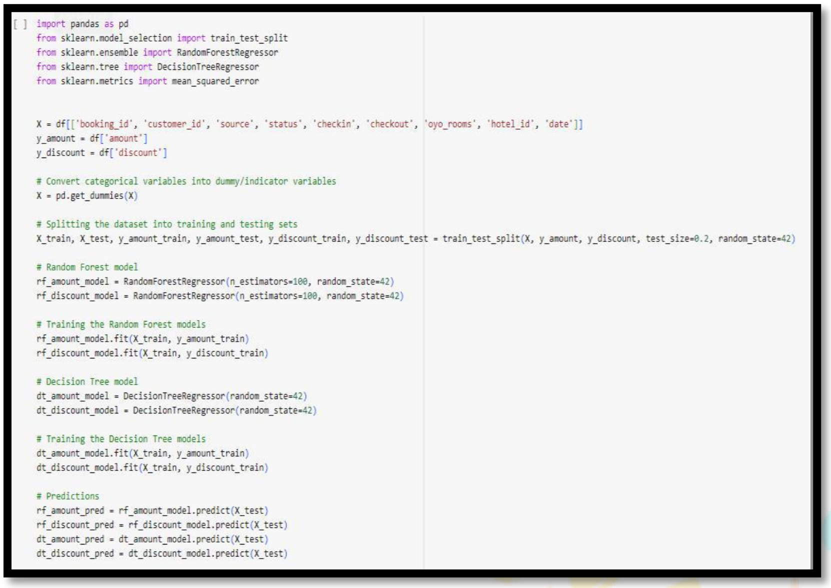


```
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, [illegible]
 0, 0, 0, 0\n0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 1, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.9779856900989544, 0.27380952380952384, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, [illegible]
 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(0.9867914144193726, 0.27380952380952384, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, [illegible]
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 1, 0, 0, 0, 0, 0, 0, 0]')]
```

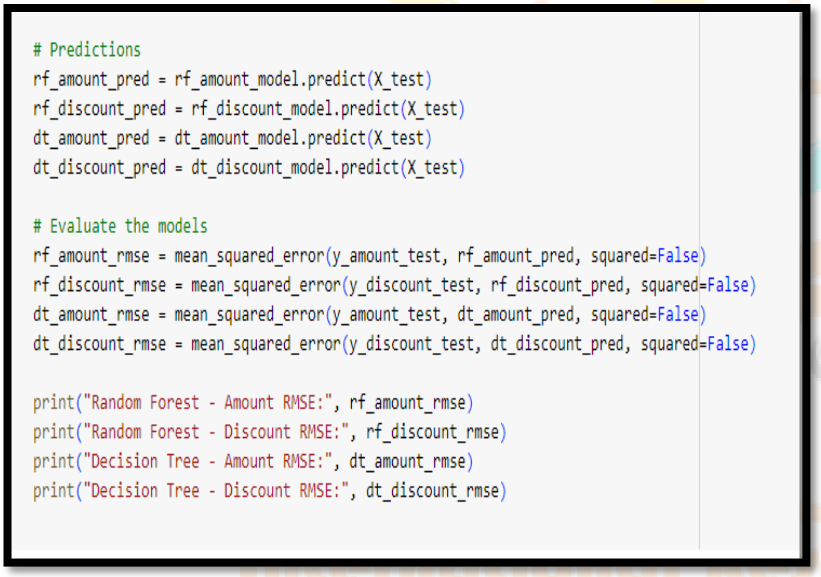


```
] model.score(X_test,y_test)

  0.65
```

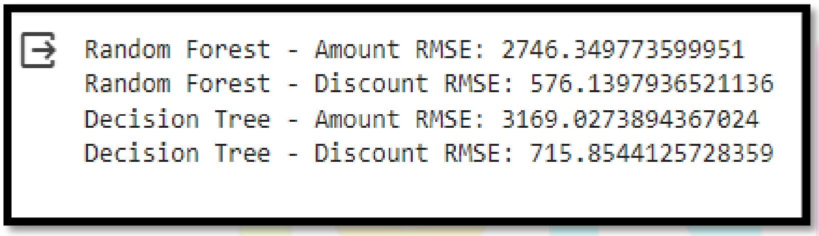
Prediction of price and discount using random forest and decision tree(code & outputs)



```
[ ] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.metrics import mean_squared_error


    X = df[['booking_id', 'customer_id', 'source', 'status', 'checkin', 'checkout', 'oyo_rooms', 'hotel_id', 'date']]
    y_amount = df['amount']
    y_discount = df['discount']

    # Convert categorical variables into dummy/indicator variables
    X = pd.get_dummies(X)

    # Splitting the dataset into training and testing sets
    X_train, X_test, y_amount_train, y_amount_test, y_discount_train, y_discount_test = train_test_split(X, y_amount, y_discount, test_size=0.2, random_state=42)

    # Random Forest model
    rf_amount_model = RandomForestRegressor(n_estimators=100, random_state=42)
    rf_discount_model = RandomForestRegressor(n_estimators=100, random_state=42)

    # Training the Random Forest models
    rf_amount_model.fit(X_train, y_amount_train)
    rf_discount_model.fit(X_train, y_discount_train)

    # Decision Tree model
    dt_amount_model = DecisionTreeRegressor(random_state=42)
    dt_discount_model = DecisionTreeRegressor(random_state=42)

    # Training the Decision Tree models
    dt_amount_model.fit(X_train, y_amount_train)
    dt_discount_model.fit(X_train, y_discount_train)

    # Predictions
    rf_amount_pred = rf_amount_model.predict(X_test)
    rf_discount_pred = rf_discount_model.predict(X_test)
    dt_amount_pred = dt_amount_model.predict(X_test)
    dt_discount_pred = dt_discount_model.predict(X_test)
```



```
# Predictions
rf_amount_pred = rf_amount_model.predict(X_test)
rf_discount_pred = rf_discount_model.predict(X_test)
dt_amount_pred = dt_amount_model.predict(X_test)
dt_discount_pred = dt_discount_model.predict(X_test)

# Evaluate the models
rf_amount_rmse = mean_squared_error(y_amount_test, rf_amount_pred, squared=False)
rf_discount_rmse = mean_squared_error(y_discount_test, rf_discount_pred, squared=False)
dt_amount_rmse = mean_squared_error(y_amount_test, dt_amount_pred, squared=False)
dt_discount_rmse = mean_squared_error(y_discount_test, dt_discount_pred, squared=False)

print("Random Forest - Amount RMSE:", rf_amount_rmse)
print("Random Forest - Discount RMSE:", rf_discount_rmse)
print("Decision Tree - Amount RMSE:", dt_amount_rmse)
print("Decision Tree - Discount RMSE:", dt_discount_rmse)
```



```
Random Forest - Amount RMSE: 2746.349773599951
Random Forest - Discount RMSE: 576.1397936521136
Decision Tree - Amount RMSE: 3169.0273894367024
Decision Tree - Discount RMSE: 715.8544125728359
```

The prediction model depicts the amount and discount that will be offered on a given random value generated using pd.get_dummies function. The model predicts the prices Ru.2746.349 and Ru.3169.027 using random forest and decision tree respectively & discounts as Ru. 576.139 and Ru. 715.854.

In conclusion, the accuracy scores obtained from testing the random forest and decision tree classifiers provide insights into their performance in classifying the target variable. While the decision tree classifier achieved higher accuracy than the random forest classifier, further analysis and refinement are necessary to optimize classifier performance and ensure robustness in real-world applications.

Each finding provides valuable insights into the effectiveness of dynamic pricing strategies, the implications of machine learning algorithms, the variability in algorithm effectiveness, and the performance of classification algorithms in predicting customer behaviour or booking status. By understanding and elaborating on these findings, businesses can make informed decisions and strategies to optimize pricing, maximize profits, and enhance overall performance in the competitive marketplace.

4. Conclusion

Dynamic pricing is a strategy employed by businesses to adjust prices of products or services in real-time, considering factors such as demand and competition. Leveraging machine learning, companies can effectively implement dynamic pricing using sophisticated algorithms. These algorithms analyze vast datasets encompassing historical sales, customer demographics, and market trends to accurately predict pricing impacts. Continuously monitoring data, machine learning models dynamically adjust prices to maximize profitability, enabling businesses to swiftly adapt to market fluctuations and tailor pricing for individual customers. This personalized approach not only boosts conversion rates and customer loyalty but also drives growth and profitability. In essence, dynamic pricing powered by machine learning offers a competitive advantage in today's data-driven marketplace by optimizing pricing strategies, enhancing revenue, and improving customer satisfaction.