Prediction of Hotel Booking Cancellation

By Group 5

Abdelrahman Ali

Esraa Fayad

Aya Metwally

University of Ottawa

DR. Olubisi Runsewe

Report structure Agenda:

Contents

I. Abstract: 3

II. Introduction: 3

IV. Model 8

V. Performance Evaluation 13

VI. Summary & Conclusion 14

VII. Bibliography 15

I. Abstract:

The hotel reservations classification problem is the task of predicting whether a hotel booking will be canceled or not canceled. This is a challenging problem because many factors can influence a customer's decision to cancel, such as the customer's travel plans, the hotel's pricing, and the customer's satisfaction with the hotel.

The Hotel Reservations Classification Dataset on Kaggle is a valuable resource for anyone who wants to learn more about this problem. The dataset contains information about 119,380 hotel bookings, including the customer's arrival date, the number of nights they stayed, the type of room they reserved, and whether or not they booked a meal plan.

The dataset has been used for a variety of research purposes, including predicting hotel cancellations, understanding customer behavior, and optimizing hotel pricing.

Now we have done our analysis to handle this problem effectively with high accuracy score in this report we will try to give details information about the problem and how we provide a model for predicting the cancelation so that the hotel owners will have more chance to reduce their losses.

II. Introduction:

The "Prediction of Hotel Booking Cancellation" system aims to address the challenges faced by hotels due to booking cancellations. Hotel industries often struggle to manage revenue and resources efficiently, leading to financial losses and customer dissatisfaction. To overcome this issue, we propose a predictive model that can forecast hotel reservation cancellations, allowing hotel managers to make informed decisions and optimize resource allocation. The hotel reservations classification problem is important for hotels because it can help them to optimize their pricing and minimize cancellations. By understanding the factors that are most likely to lead to a cancellation, hotels can adjust their pricing accordingly. For example, if a hotel knows that customers are more likely to cancel their bookings if the price is too high, they can lower their prices to reduce the number of cancellations.

First, we get a description of the columns that we have in the dataset and we get information for each feature

```
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  object
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  object
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  object
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  object
dtypes: float64(1), int64(13), object(5)
```

After that, we started to visualize each feature to get the best description of it using Dataprep and writing our code as well.

Then we do some data preprocessing for each feature such as:

- Check if there are duplicates
- Check if there are missing values
- Display data types about features
- Check for outliers
- Get unique values
- Visualize missing values
- Find the relationship between features
- Compute the correlation matrix
- Show the distribution of most of the columns and handle outliers by log normalization and calculate the probabilities in some of them

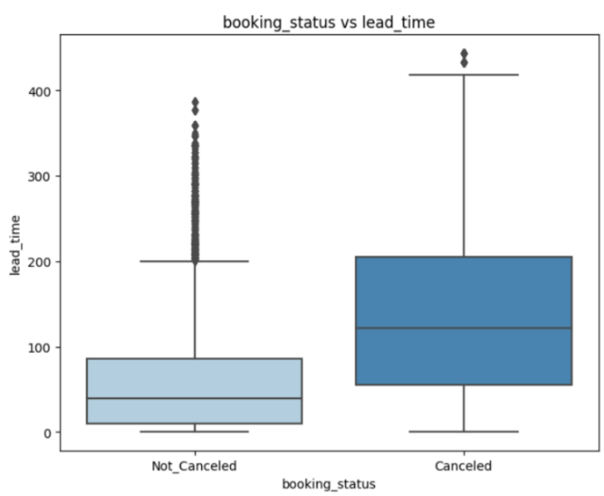
Check missing and duplicate rows in the dataset:

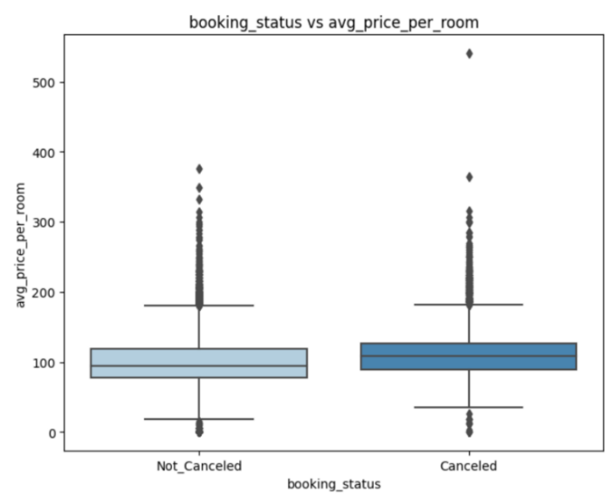
Dataset Statistics

Number of Variables	19
Number of Rows	36275
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	15.4 MB
Average Row Size in Memory	444.4 B
Variable Types	Categorical: 13 Numerical: 6

We found that the dataset is clean and only needs some transformation.

After that, we display Boxplots for all numeric features and histograms for categorical features. For example, 'lead_time' and 'avg_price_per_room' Features against our target column booking status.





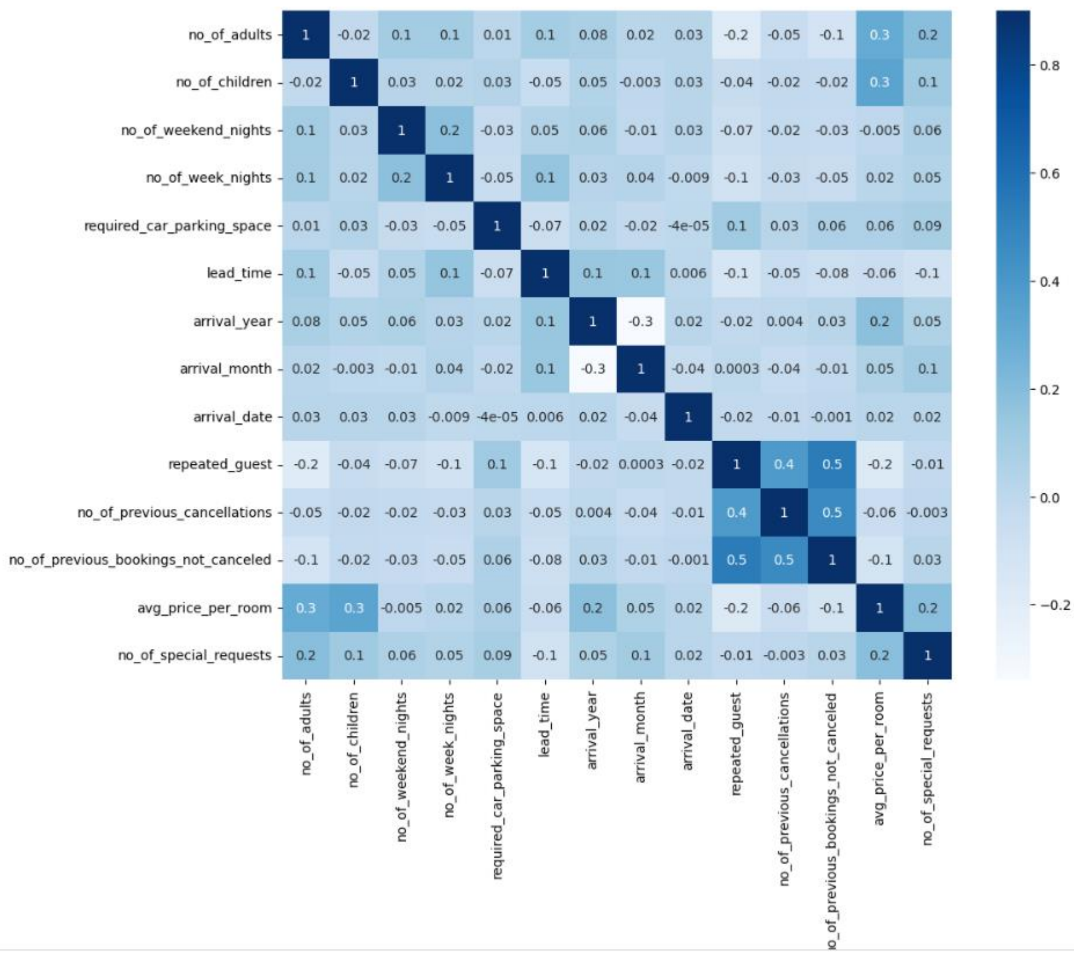
This Chart indicates that if the difference between the booking date and arrival date we noticed that more than 90 days almost will lead to cancellation but if the booking is near arrival this will lead to no cancellation of the booking.

We gain more insights from the visualizations and they help in unlocking valuable information from the data, improving model understanding, and driving better predictions and business decisions for hotel revenue management.

After that we make Feature Engineering handling correlated data features and drop correlated features if any exist these results in:

The Highly correlated features: ['no_of_previous_bookings_not_canceled'] based on threshold = 0.5 and we choose to drop it.

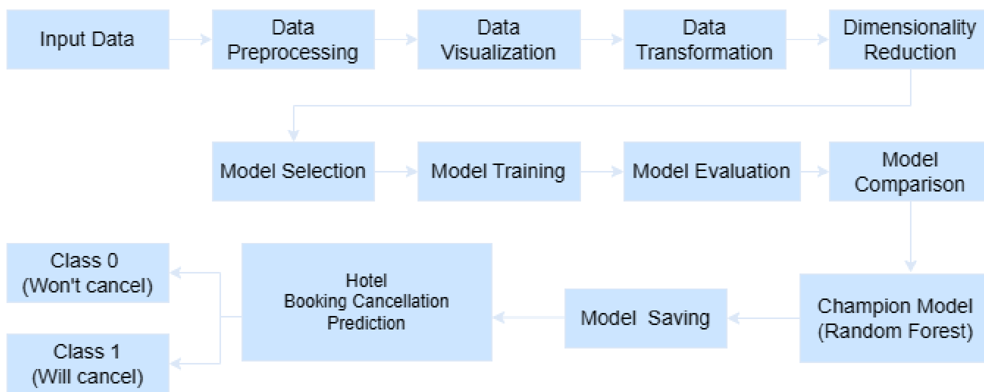
The heatmap that visualizes the relationships and patterns between two or more variables in the dataset:



III. System Architecture

The system architecture is a crucial aspect of the "Prediction of Hotel Booking Cancellation" system. A well-designed architecture ensures that the prediction model is accurate, scalable, and easy to maintain. By establishing a clear blueprint for the system's structure, we can identify potential bottlenecks, plan for future expansion, and implement the best machine-learning algorithms to achieve high accuracy.

The following diagram illustrates the high-level system architecture for the "Prediction of Hotel Booking Cancellation" system:

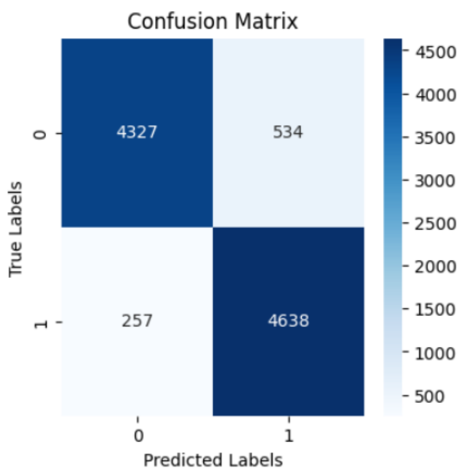


In the diagram, we showcase the flow of data and the interactions between different components, including data collection, preprocessing, visualization, transformation, model training, and model deployment. The diagram provides a visual representation of the system's design and helps stakeholders understand the overall structure and functionality of the "Prediction of Hotel Booking Cancellation" system.

IV. Model

First of all, we spilt our data into training and testing sets then we start to use our models:

Decision Tree: we applied it and here are our analysis and results:

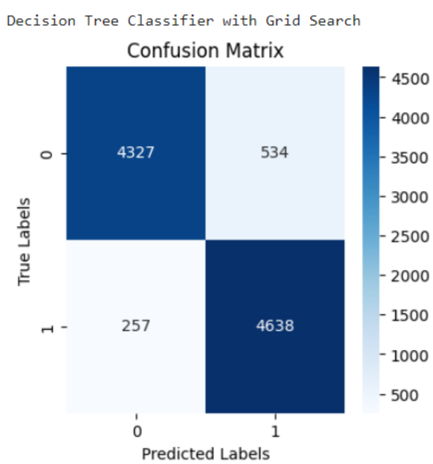



```
                 Predicted Labels
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.89      0.92      4861
           1       0.90      0.95      0.92      4895

    accuracy                           0.92      9756
   macro avg       0.92      0.92      0.92      9756
weighted avg       0.92      0.92      0.92      9756

Training Accuracy of Decision Tree: 0.9939011890118902
Testing Accurcy of Decision Tree: 0.9189216892168922
```

The first figure represents the confusion matrix Second figure represents the results of the Classification Report

The accuracy of mode for training = 99.3% and For Test Accuracy = 91.8%

Then we applied grid search for the Decision tree:



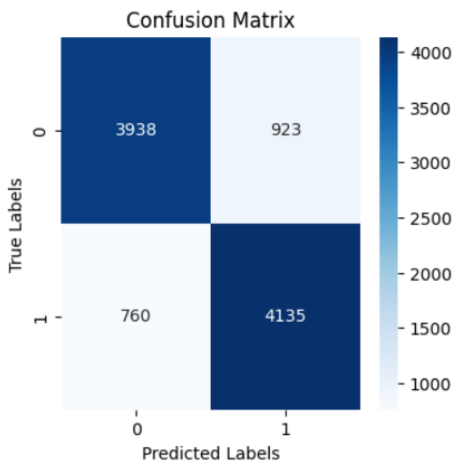

```
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.89      0.92      4861
           1       0.90      0.95      0.92      4895

    accuracy                           0.92      9756
   macro avg       0.92      0.92      0.92      9756
weighted avg       0.92      0.92      0.92      9756

Training Accuracy of Grid Decision Tree: 0.8935783107831078
Accurcy of Grid Decision Tree: 0.9189216892168922
```

The first figure represents the confusion matrix Second figure represents the results of the Classification Report

The accuracy of mode for training = 89.3% and For Test Accuracy = 91.8%

Support Vector Machine Classifier:

```
Support Vector Machine Classifier:
```

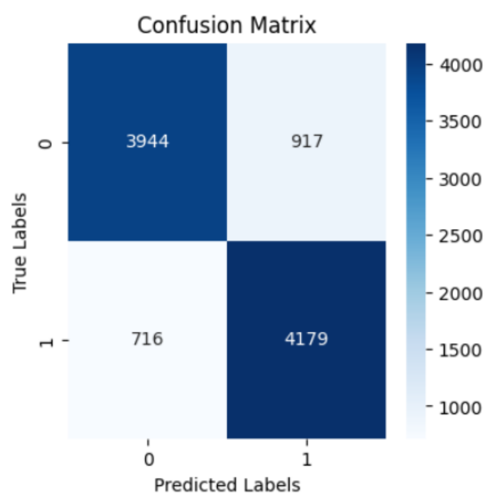



```
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.81      0.82      4861
           1       0.82      0.84      0.83      4895

    accuracy                           0.83      9756
   macro avg       0.83      0.83      0.83      9756
weighted avg       0.83      0.83      0.83      9756

Training Accuracy of SVM: 0.8320520705207052
Testing Accurcy of SVM: 0.827490774907749
```

The first figure represents the confusion matrix Second figure represents the results of the Classification Report

The accuracy of mode for training = 83.2% and For Test Accuracy = 82.7%

Then we applied grid search for Support Vector Machine:



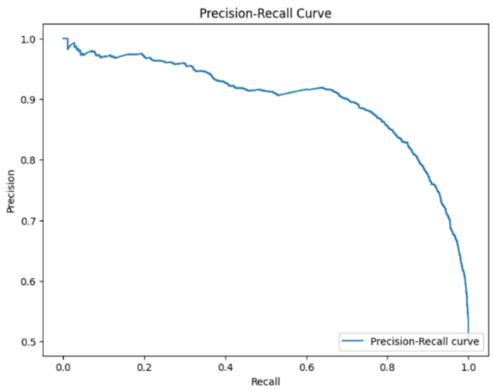

```
Support Vector Machine Classifier with Grid Search:
```

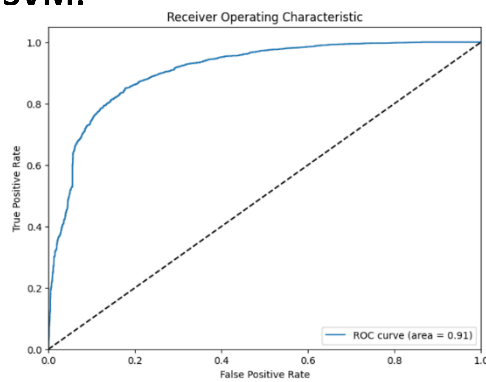
```
Predicted Labels
Support Vector Machine Classifier with Grid Search:
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.81      0.83      4861
           1       0.82      0.85      0.84      4895

    accuracy                           0.83      9756
   macro avg       0.83      0.83      0.83      9756
weighted avg       0.83      0.83      0.83      9756

Training Accuracy of Grid SVM: 0.8414052890528906
Testing Accurcy of Grid SVM: 0.8326158261582616
```

The first figure represents the confusion matrix Second figure represents the results of the Classification Report

The accuracy of mode for training = 84.1% and For Test Accuracy = 83.2%

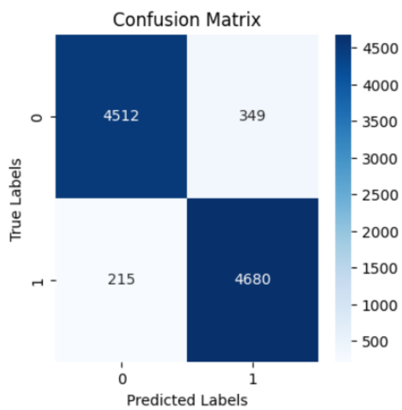
Precision-Recall curve and ROC Curve for SVM:





Random Forest:

Random Forest Classifier



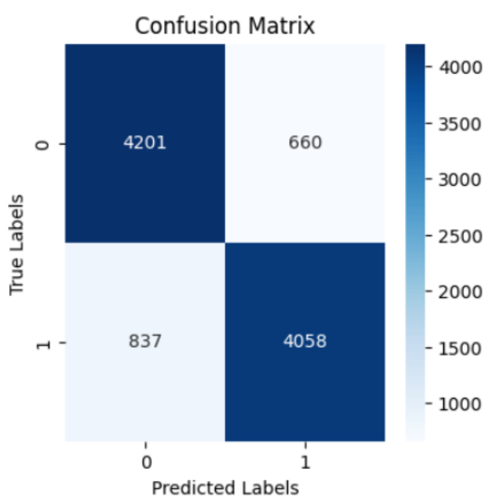
Predicted Labels

```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      4861
           1       0.93      0.96      0.94      4895

    accuracy                           0.94      9756
   macro avg       0.94      0.94      0.94      9756
weighted avg       0.94      0.94      0.94      9756

Training Accuracy of Random Forest: 0.9939011890118902
Testing Accurcy of Random Forest: 0.942189421894219
```

The accuracy of mode for training = 99.3% and For Test Accuracy = 94.2%

Random Forest with grid search:

Random Forest Classifier with Grid Search



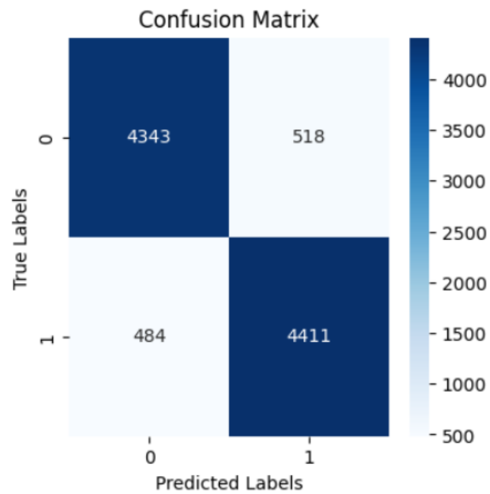
Predicted Labels

```
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85      4861
           1       0.86      0.83      0.84      4895

    accuracy                           0.85      9756
   macro avg       0.85      0.85      0.85      9756
weighted avg       0.85      0.85      0.85      9756

Training Accuracy of Grid Random Forest: 0.8440959409594095
Testing Accurcy of Grid Random Forest: 0.8465559655596556
```

The accuracy of mode for training = 84.4% and For Test Accuracy = 84.6%

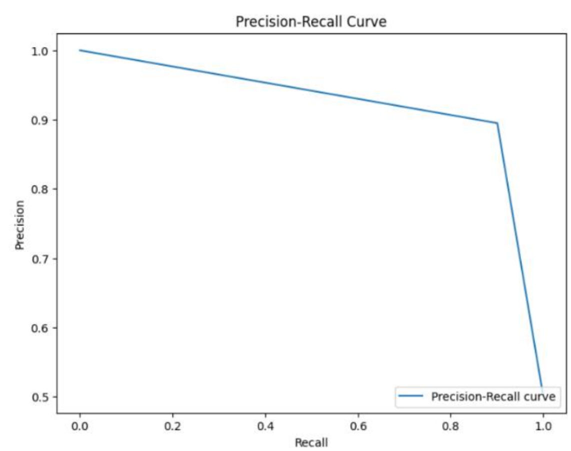
XG-Boost:

```
XGBoost Classifier:
```



Predicted Labels

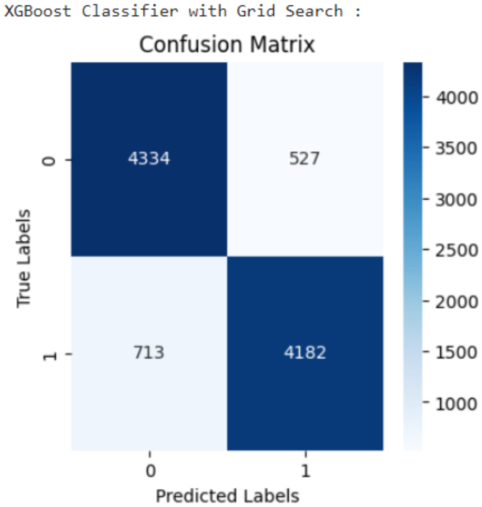
```
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.89      0.90      4861
           1       0.89      0.90      0.90      4895

    accuracy                           0.90      9756
   macro avg       0.90      0.90      0.90      9756
weighted avg       0.90      0.90      0.90      9756

Training Accuracy of XGBoost: 0.9198185731857319
Testing Accurcy of XGBoost: 0.8972939729397293
```



The accuracy of mode for training = 91.9% and For Test Accuracy = 89.7%

XG-Boost with grid search:

```
XGBoost Classifier with Grid Search :
```

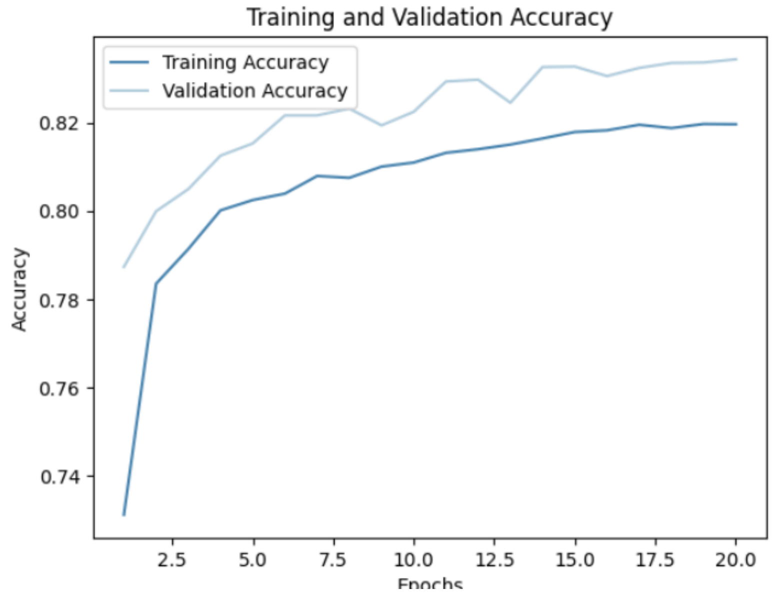


Predicted Labels

```
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.89      0.87      4861
           1       0.89      0.85      0.87      4895

    accuracy                           0.87      9756
   macro avg       0.87      0.87      0.87      9756
weighted avg       0.87      0.87      0.87      9756

Training Accuracy of Grid Random Forest: 0.8753587535875359
Testing Accurcy of Grid XGBoost: 0.8728987289872898
```

The accuracy of mode for training = 87.5% and For Test Accuracy = 87.2%

Neural network:

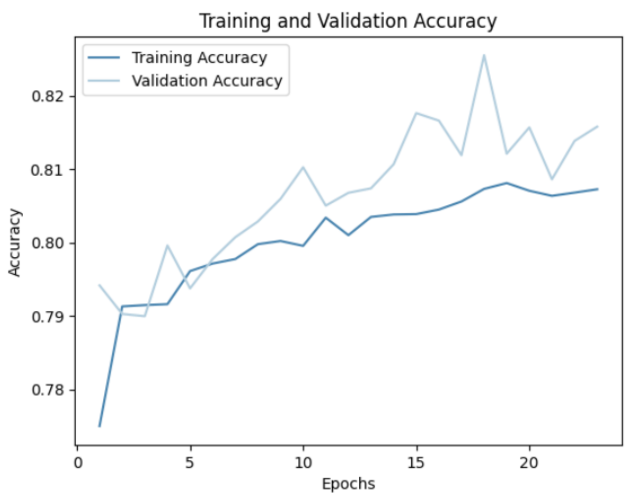


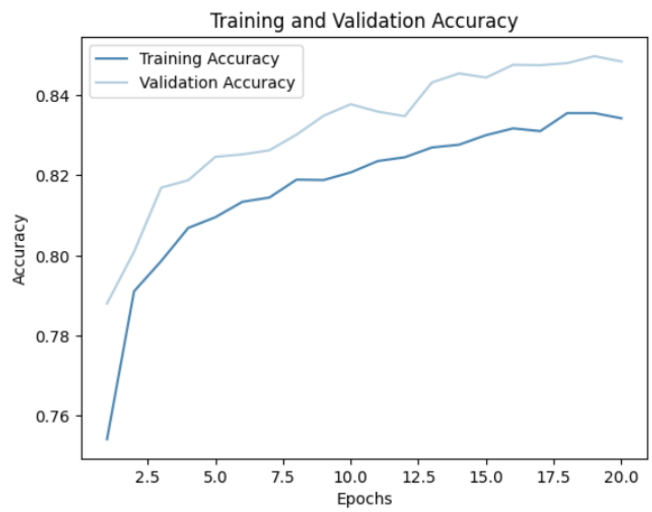

```
1220/1220 [==============================] - 3s 2ms/step - loss: 0.3626 - accuracy: 0.8348
Training Accuracy of Neural Network Model: 0.8348196148872375
305/305 [==============================] - 1s 2ms/step - loss: 0.3717 - accuracy: 0.8344
Testing Accuracy of Neural Network Model:: 0.8343583345413208
```

The accuracy of mode for training = 83.4% and For Test Accuracy = 83.4%

Neural network with grid search:

```
Best Learning Rate: 0.001
Best Dropout Rate: 0.3
```

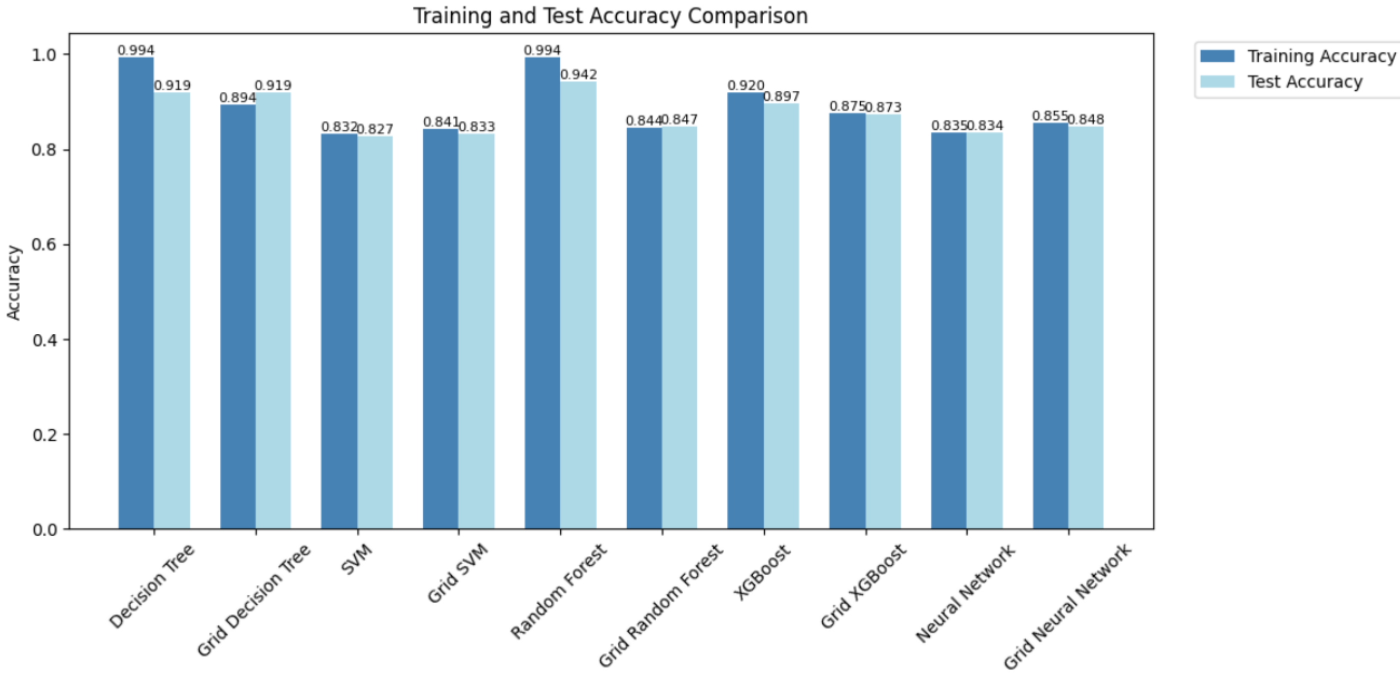





```
1220/1220 [==============================] - 3s 2ms/step - loss: 0.3363 - accuracy: 0.8545
Training Accuracy of Neural Network Model: 0.8348196148872375
305/305 [==============================] - 1s 3ms/step - loss: 0.3461 - accuracy: 0.8484
Testing Accuracy of Neural Network Model:: 0.8343583345413208
```

The accuracy of mode for training = 85.4% and For Test Accuracy = 84.8%

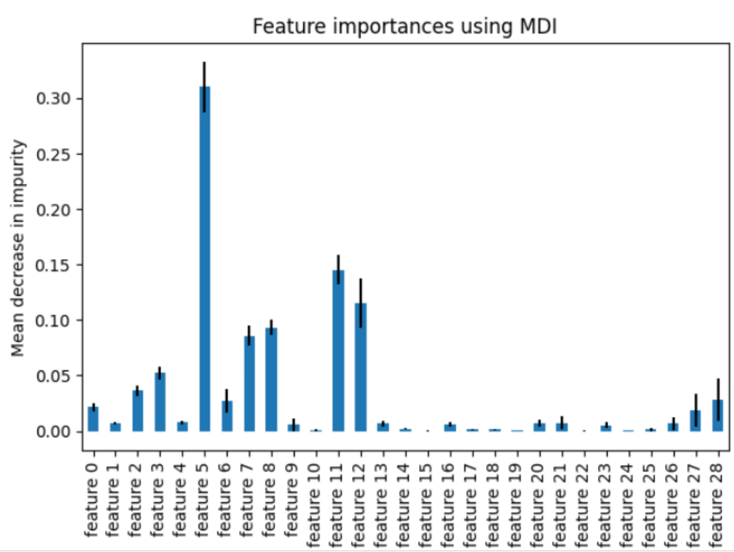
V. Performance Evaluation



We make a plot to show the accuracy of all models in the training and testing dataset.

As we can see, the best model based on testing accuracy is the Random Forest algorithm. The reason why is neural network doesn't achieve the highest score is that our data is tabular and small, so it is not the best technique for it. The ML model is best for our case. Neural networks typically require a large amount of data to learn complex patterns effectively. As in our project, If the dataset is small, other algorithms like decision trees or random forests might generalize better with limited data.

We plot feature importance using MDI to help us understand which features have the most significant impact on the model's predictions and found that feature 5 which refers to “lead_time” is the most important in our dataset.



VI. Summary & Conclusion

We can conclude that the hotel reservations classification problem can be effectively addressed using machine learning models. The best-performing model on the testing dataset is determined to be **Random Forest** with a score = **94.2 %,** achieving a high accuracy score. Neural Networks, while having the potential for complex patterns, did not perform as well in this specific case due to the limited amount of tabular data available.

Although we did hyperparameter tuning for improving the performance using grid search for our five models, the accuracy didn't increase to the accuracy of the model with default parameters. The reason for that is hyperparameter tuning can significantly impact the model's performance, but finding the right combination of hyperparameters often requires experimentation, patience, and a good understanding of the problem and the algorithms used.

By leveraging the insights gained from the models and feature importance analysis, hotel owners can optimize their pricing strategies and minimize cancellations. Understanding the factors that contribute most to cancellations can help hotels adjust their pricing and other policies to improve customer satisfaction and reduce losses.

VII. Bibliography

Dataset link: **https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset**

Notebook link:
https://colab.research.google.com/drive/1v0aZBO3AMP5PNynN99Z0N7JWKYZuxGfq?usp=sharing#scrollTo=9J-kooD8mfFk

References:

1. **Banza, M. (n.d.). Predicting Hotel Booking Cancellations Using Machine Learning. Step-by-Step Guide with Real Data and Python. https://www.linkedin.com/pulse/u-hotel-booking-cancellations-using-machine-learning-manuel-banza/**
2. **Raza, A. (2023) Hotel Reservations dataset, Kaggle. Available at: https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset (Accessed: 17 July 2023).**
3. **Eleazar C-Sánchez, A. J. Sánchez-Medina, and L. Romero-Domínguez, "Forecasting Hotel-booking Cancelations Using Personal Name Records: An Artificial Intelligence Approach," pp. 3–14, Jan. 2022, doi: https://doi.org/10.1007/978-981-16-9268-0_1.**