Hotel Cancellation Predictions

Project 4 - Group 9

Hampton Hughes, Andrew Grimm, Fekadu Habeteyohannes, Brielle Bernardy, Karen Fan

Introduction

Online hotel reservations channels have changed booking possibilities making it more convenient for customers to book, make adjustments and also to cancel reservations when necessary.

- Average percentage of cancellations across all sources is approx 24%; solely online booking cancellations is 38% while offline cancellation rates is at 10%
- Online reservation cancellation breakdowns for common online booking sites-
  - Booking.com cancellation avg at 57%
  - Expedia cancellation avg at 26%
  - Hotel official websites cancellation avg at 14%

Typical reasons for cancellations are due to change of plans/ scheduling conflicts, and customers are more likely to cancel if it’s of no cost or if charges are low. This leads to lost of revenue for hotels especially if cancellations are done last minute.

Our project aims to predict the likeliness of hotel cancellations prior to trip start date.

Dataset Review

In Machine Learning- this is a Classification Problem

Hotel Reservations Dataset from Kaggle, sourced from real hotels in Portugal with PII removed

- Target= booking_status

- Features = no_of_adults, no_of_children, no_of_weekend_nights, no_of_week_nights, type_of_meal_plan, required_car_parking_space, room_type_reserved, lead_time, arrival_month, market_segment_type, repeated_guest, no_of_previous_cancellations, no_of_previous_bookings_not_canceled, avg_price_per_room, no_of_special_requests

Tools/ Metrics Utilized

Metrics to Evaluate Model

- Classification Report
- Confusion Matrix

Classification Models used

- Random Forest
- KNN

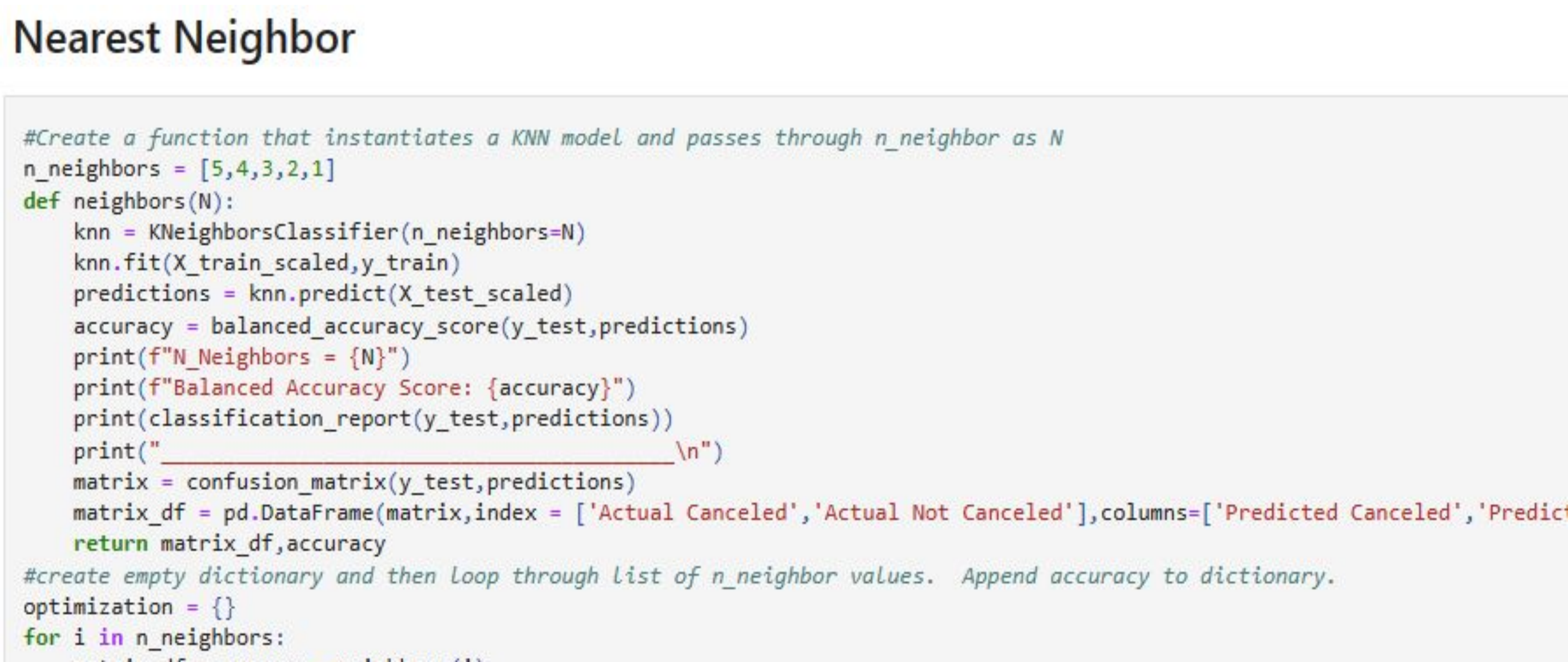
Visualization Tool

- Matplotlib

Tools used to Transform/ Prep Data

- Pandas
- Scikit Learn
- Scikit Standard Scalar
- Get Dummies
- Sqlite

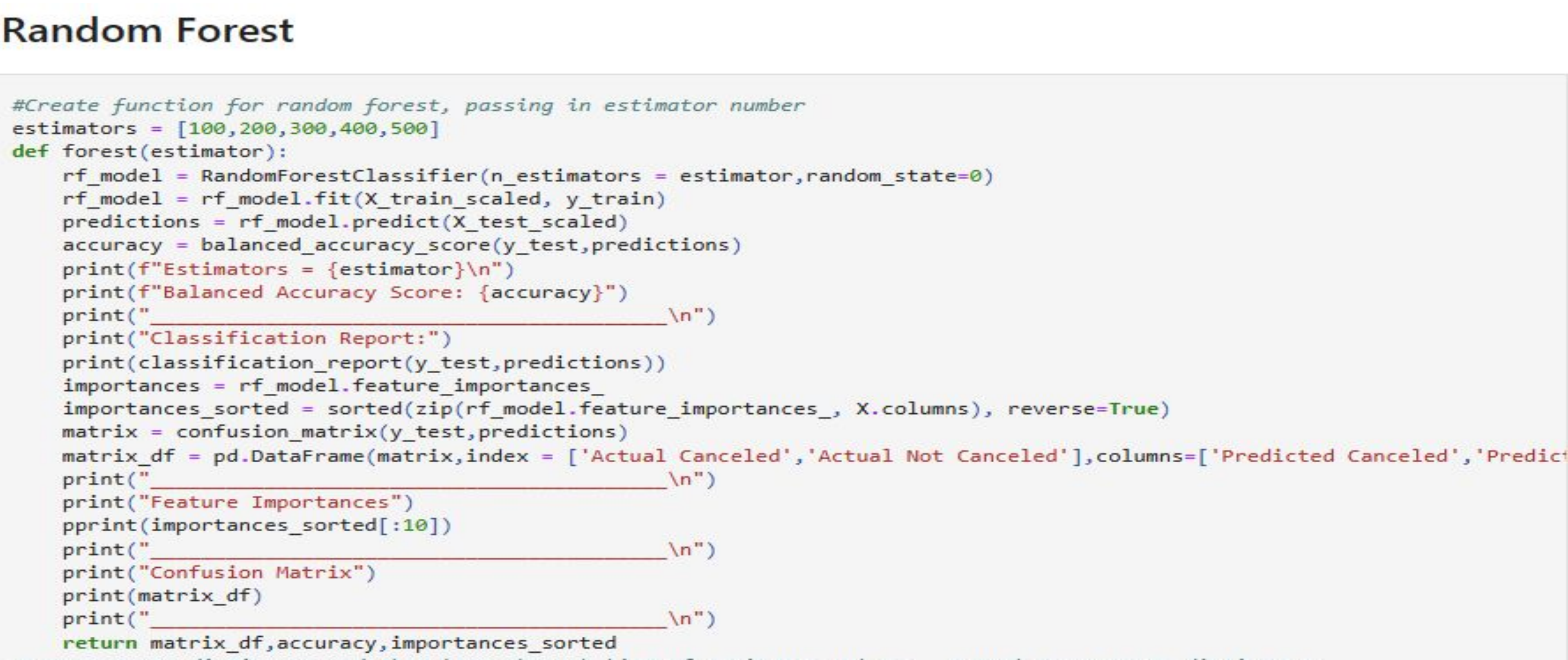
Function for K Nearest Neighbors model



Nearest Neighbor

```
#Create a function that instantiates a KNN model and passes through n_neighbor as N
n_neighbors = [5,4,3,2,1]
def neighbors(N):
    knn = KNeighborsClassifier(n_neighbors=N)
    knn.fit(X_train_scaled,y_train)
    predictions = knn.predict(X_test_scaled)
    accuracy = balanced_accuracy_score(y_test,predictions)
    print(f"N_Neighbors = {N}")
    print(f"Balanced Accuracy Score: {accuracy}")
    print(classification_report(y_test,predictions))
    print("_________________________________________\n")
    matrix = confusion_matrix(y_test,predictions)
    matrix_df = pd.DataFrame(matrix,index = ['Actual Canceled','Actual Not Canceled'],columns=['Predicted Canceled','Predic
    return matrix_df,accuracy
#create empty dictionary and then loop through list of n_neighbor values.  Append accuracy to dictionary.
optimization = {}
for i in n_neighbors:
```

Function for Random Forest model

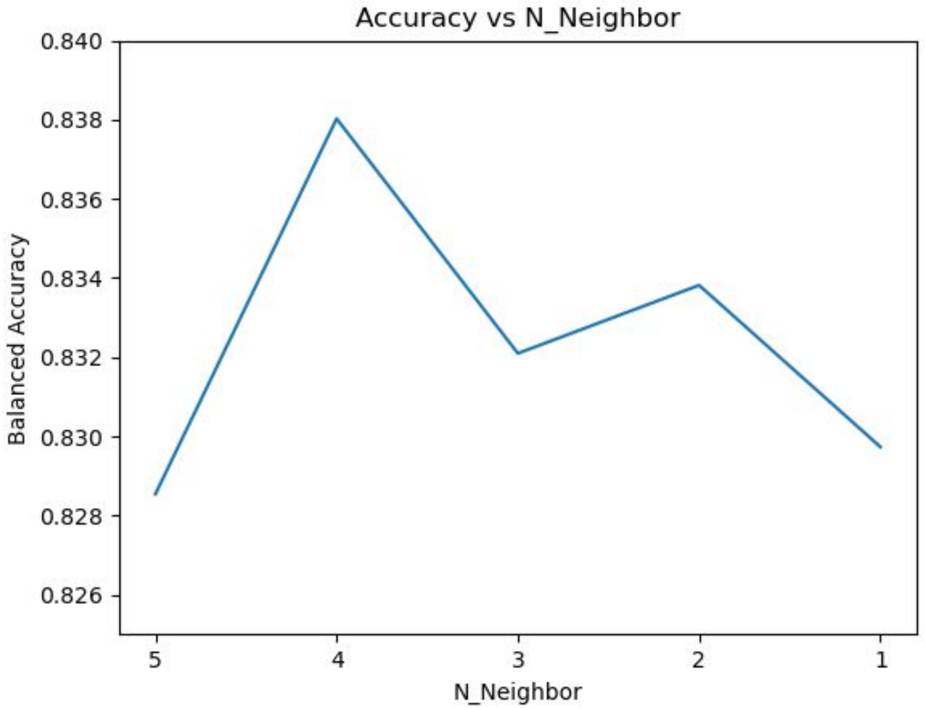


Random Forest

```
#Create function for random forest, passing in estimator number
estimators = [100,200,300,400,500]
def forest(estimator):
    rf_model = RandomForestClassifier(n_estimators = estimator,random_state=0)
    rf_model = rf_model.fit(X_train_scaled, y_train)
    predictions = rf_model.predict(X_test_scaled)
    accuracy = balanced_accuracy_score(y_test,predictions)
    print(f"Estimators = {estimator}\n")
    print(f"Balanced Accuracy Score: {accuracy}")
    print("________________________________________\n")
    print("Classification Report:")
    print(classification_report(y_test,predictions))
    importances = rf_model.feature_importances_
    importances_sorted = sorted(zip(rf_model.feature_importances_, X.columns), reverse=True)
    matrix = confusion_matrix(y_test,predictions)
    matrix_df = pd.DataFrame(matrix,index = ['Actual Canceled','Actual Not Canceled'],columns=['Predicted Canceled','Predict
    print("________________________________________\n")
    print("Feature Importances")
    pprint(importances_sorted[:10])
    print("________________________________________\n")
    print("Confusion Matrix")
    print(matrix_df)
    print("________________________________________\n")
    return matrix_df,accuracy,importances_sorted
```

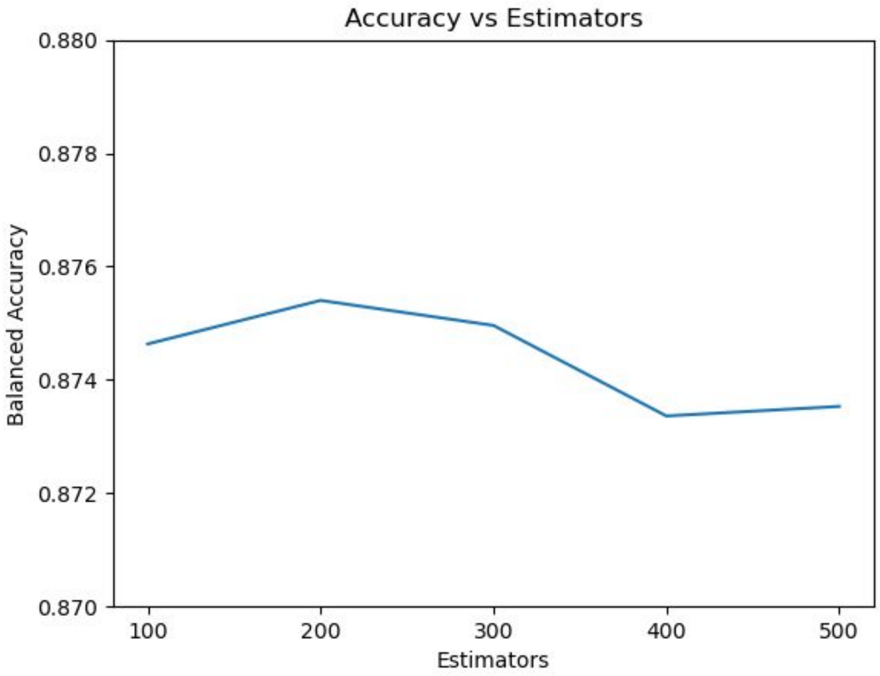
K Nearest Neighbors Findings

- Balanced accuracy is fairly steady across different values of n_neighbor
- With n_neighbors = 4, the KNN model achieves a balanced accuracy of 83.8%



Random Forest Findings

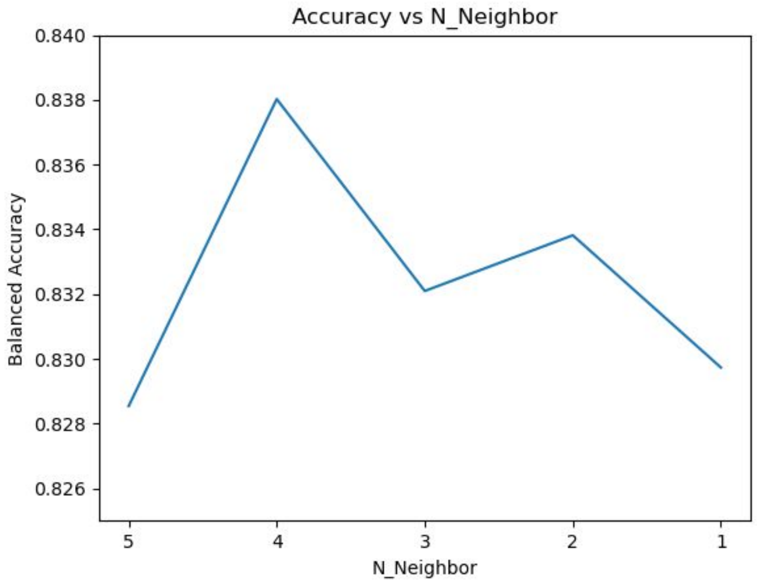
- Changing the number of estimators has a small impact on model performance
- With 200 estimators, the random forest model achieves a balanced accuracy of 87.5% – the highest out of all of our models

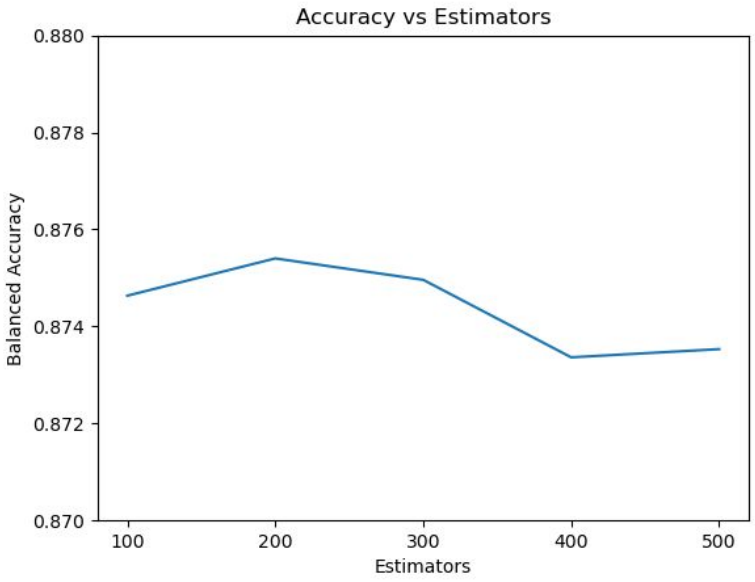


Classification Model Comparison

K Nearest Neighbor Model

Random Forest Model

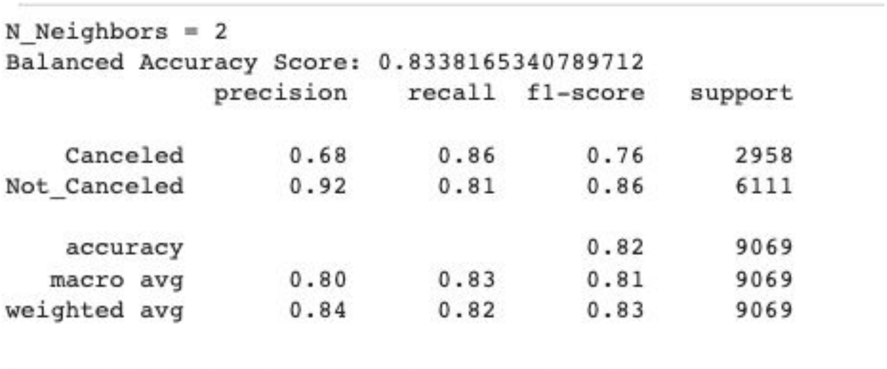




Precision/Recall comparison

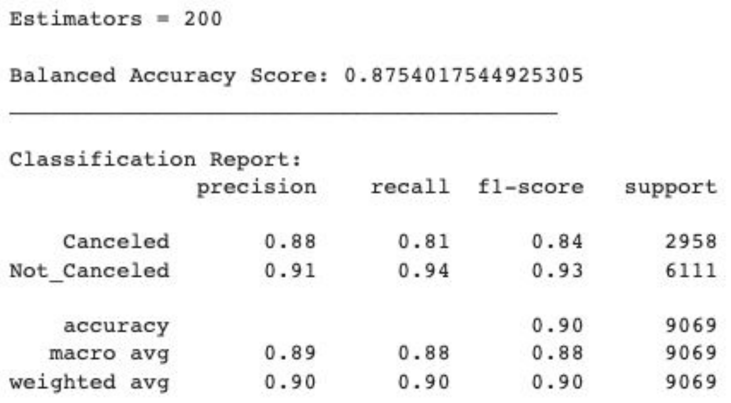
- We could choose to optimize for a higher precision or recall
- For example, if we wanted to capture more of the canceled reservations, we could use the Nearest Neighbor model that yields a recall of 86%
- On the downside, we would also be misclassifying a higher number of the non-canceled reservations

Nearest Neighbor



```
N_Neighbors = 2
Balanced Accuracy Score: 0.8338165340789712
              precision    recall  f1-score   support

    Canceled       0.68      0.86      0.76      2958
Not_Canceled       0.92      0.81      0.86      6111

    accuracy                           0.82      9069
   macro avg       0.80      0.83      0.81      9069
weighted avg       0.84      0.82      0.83      9069
```

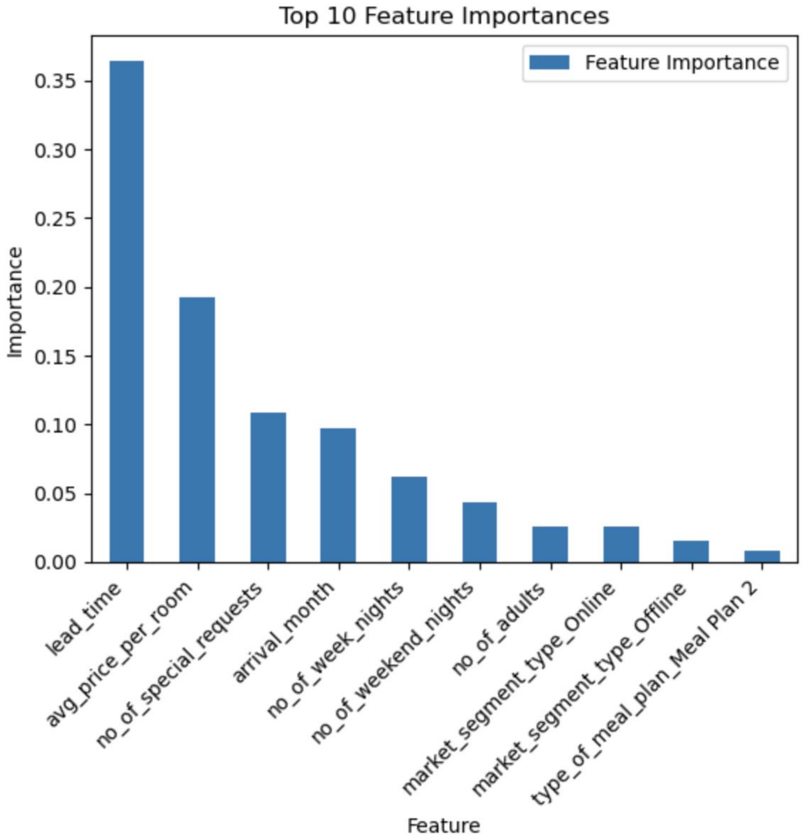
Random Forest



```
Estimators = 200

Balanced Accuracy Score: 0.8754017544925305
_____________________________________________

Classification Report:
              precision    recall  f1-score   support

    Canceled       0.88      0.81      0.84      2958
Not_Canceled       0.91      0.94      0.93      6111

    accuracy                           0.90      9069
   macro avg       0.89      0.88      0.88      9069
weighted avg       0.90      0.90      0.90      9069
```

Most Important Features in Predicting Cancellations

The most important feature in predicting whether or not someone will cancel their hotel reservation is: the lead time of the reservation, followed by the average price per room.



Thank You

Questions/ Comments?

Resources– Citations

https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset?resource=download

https://experience-crm.fr/en/where-do-cancellations-come-from/#:~:text=First%20and%20foremost%2C%20let's%20try,sources%2C%20is%20currently%2024%25.