DSC 630 Milestone 4: Finalizing Your Results

Chitramoy Mukherjee

Predictive analysis on Hotel Booking Cancellation using

hotel_bookings.csv dataset

Data Preparation:

Data preparation is the first and most important step of model building. By following these data preparation steps, we can ensure that the dataset is clean, relevant, and properly formatted for training a Hotel Booking Cancellation Prediction Model. This will ultimately lead to better model performance and more accurate predictions. Dataset Basic Information is already provided in the earlier milestone where we have provided total Number of Entries, Columns, Data Types and Missing values. Most of the columns, 16 to be precise, are of the object data type and 16 columns are of the int64 data type, representing integer values and 4 columns are of the float64 data type. During data preparation followed the below steps,

1. Identify the Null values and fill null values with zero and remove the rows.
2. adults, babies and children can't be zero at same time, so dropping the rows having all these zero at same time as those are not valid bookings.
3. Split the data into test and training dataset.
4. Identify the categorical and numerical columns.

After initial data cleaning and removing unwanted columns from the dataset and after splitting the data into test and training, I am planning to execute Logistic regression, KNN, Decision Tree, Random Forest and XGBoost Modeling on the data. Will evaluate the Accuracy Score, Confusion Matrix, precession and F1-score of individual models to decide which model to be used to identify the cancellation prediction of hotel booking using the hotel_bookings dataset data.

```
# importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn.ensemble import ExtraTreesClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import VotingClassifier
import folium
from folium.plugins import HeatMap
import plotly.express as px
import sort_dataframeby_monthorweek as sd
plt.style.use('fivethirtyeight')
%matplotlib inline
pd.set_option('display.max_columns', 32)
```

```
# reading data
df = pd.read_csv('C:\\Users\\Chitramoy\\Desktop\\MS-DSC\\DSC-630\\Week-9\\hotel_bookings.csv')
df.head()

# Data Types:
# A majority of the columns, 16 to be precise, are of the object data type (often representing strings or categorical data).
# 16 columns are of the int64 data type, representing integer values.
# 4 columns are of the float64 data type, which typically denotes decimal values.
# Missing Values:
# The column children has 4 missing values.
# The column country has 488 missing values.
# The column agent has 16,340 missing values.
# The column company has a significant number of missing values, totaling 112,593

# Based on the data types and the feature explanations provided earlier, we identified that 20 columns (hotel, is_canceled, arrival_date_year, arrival_date_month,
# meal, country, market_segment, distribution_channel, is_repeated_guest, reserved_room_type, assigned_room_type, deposit_type, agent, company, customer_type,
# reservation_status, name, email, phone-number and credit_card) are categorical in terms of their semantics. These features must have string (object)
# data type to ensure proper analysis and interpretation in subsequent steps.
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment	distribution_channel	is_repeated_
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct	Direct	
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	0.0	0	BB	PRT	Direct	Direct	
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Direct	Direct	
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	0.0	0	BB	GBR	Corporate	Corporate	
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	0.0	0	BB	GBR	Online TA	TA/TO	

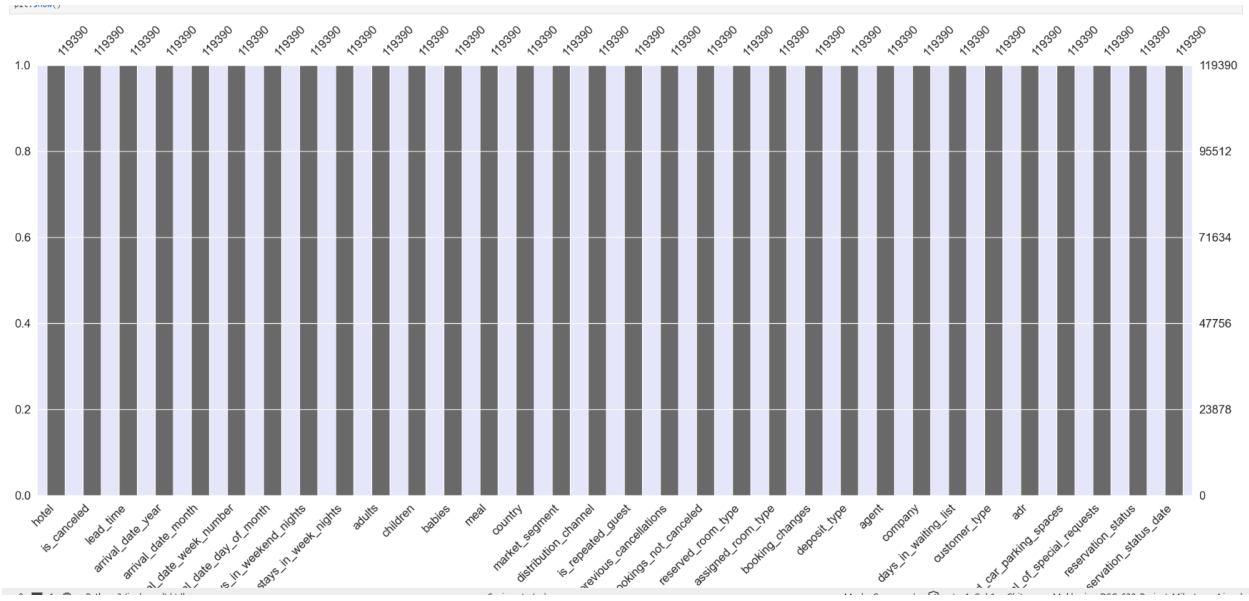
```
df.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeated_guest	previous_cancellations	previous_booking
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119386.000000	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	2.500302	1.856403	0.103890	0.007949	0.031912	0.087118	
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	1.908286	0.579261	0.398561	0.097436	0.175767	0.844336	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000	2.000000	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	3.000000	2.000000	0.000000	0.000000	0.000000	0.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000	55.000000	10.000000	10.000000	1.000000	26.000000	

```
# checking for null values in the dataset
null = pd.DataFrame({'Null Values' : df.isna().sum(), 'Percentage Null Values' : (df.isna().sum()) / (df.shape[0]) * (100)})
null
```

```
# filling null values with zero
df.fillna(0, inplace = True)
```

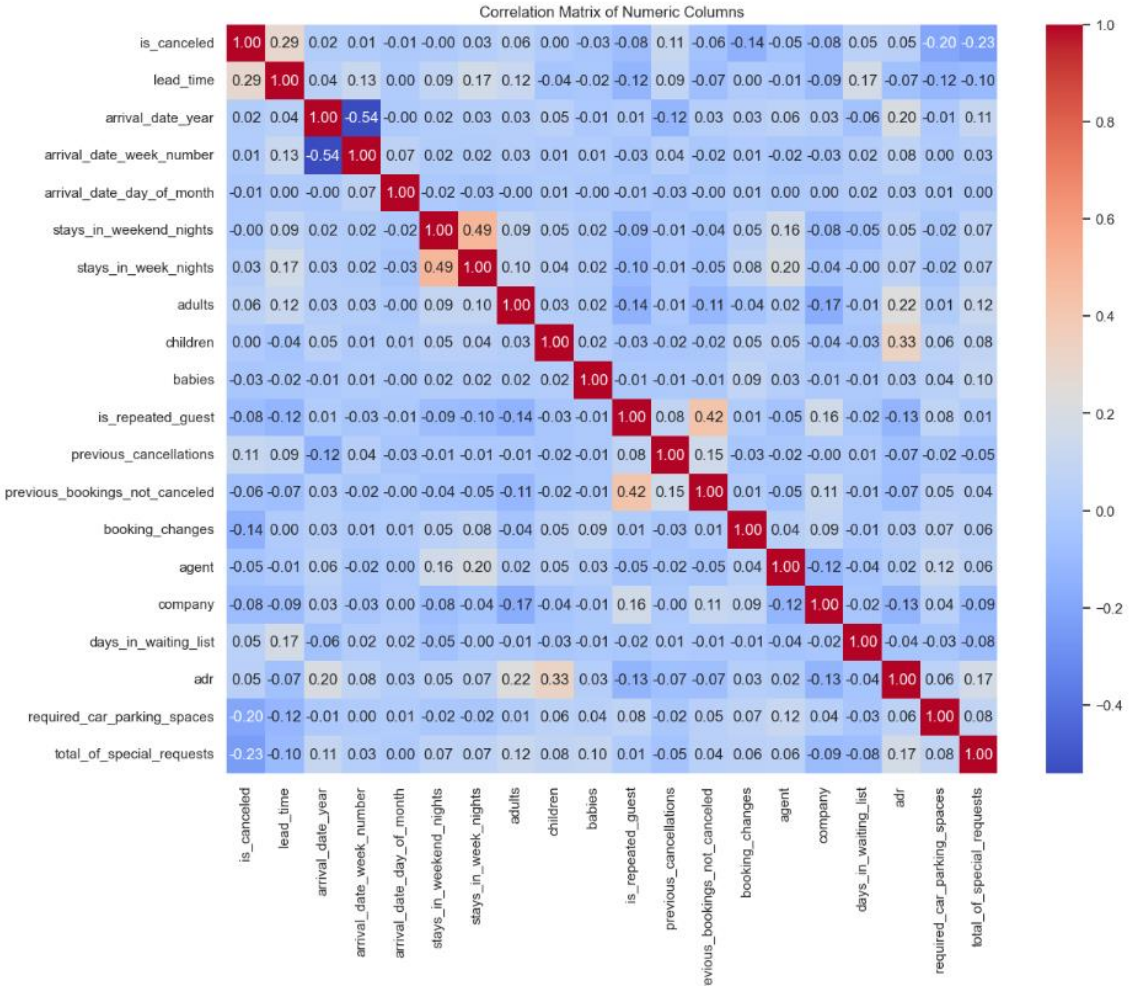
```
# visualizing null values
msno.bar(df)
plt.show()
```




```
# Identify categorical and numeric columns
numeric_columns = df.select_dtypes(include=['int64', 'float64']).columns

# Create correlation matrix for numeric columns
correlation_matrix = df[numeric_columns].corr()

# Plot the correlation matrix
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numeric Columns')
plt.show()
```



In [53]:
```
# dropping columns that are not useful

useless_col = ['days_in_waiting_list', 'arrival_date_year', 'arrival_date_year', 'assigned_room_type', 'booking_changes',
               'reservation_status', 'country', 'days_in_waiting_list']

df.drop(useless_col, axis = 1, inplace = True)
```

In [54]:
```
# creating numerical and categorical dataframes

cat_cols = [col for col in df.columns if df[col].dtype == 'O']
cat_cols
```

Out[54]:
```
['hotel',
 'arrival_date_month',
 'meal',
 'market_segment',
 'distribution_channel',
 'reserved_room_type',
 'deposit_type',
 'customer_type',
 'reservation_status_date']
```

In [55]:
```
cat_df = df[cat_cols]
cat_df.head()
```

Out[55]:

	hotel	arrival_date_month	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	reservation_status_date
0	Resort Hotel	July	BB	Direct	Direct	C	No Deposit	Transient	2015-07-01
1	Resort Hotel	July	BB	Direct	Direct	C	No Deposit	Transient	2015-07-01
2	Resort Hotel	July	BB	Direct	Direct	A	No Deposit	Transient	2015-07-02
3	Resort Hotel	July	BB	Corporate	Corporate	A	No Deposit	Transient	2015-07-02
4	Resort Hotel	July	BB	Online TA	TA/TO	A	No Deposit	Transient	2015-07-03

```
cat_df['reservation_status_date'] = pd.to_datetime(cat_df['reservation_status_date'])

cat_df['year'] = cat_df['reservation_status_date'].dt.year
cat_df['month'] = cat_df['reservation_status_date'].dt.month
cat_df['day'] = cat_df['reservation_status_date'].dt.day
```

```
cat_df.drop(['reservation_status_date','arrival_date_month'] , axis = 1, inplace = True)
```

```
cat_df.head()
```

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day
0	Resort Hotel	BB	Direct	Direct	C	No Deposit	Transient	2015	7	1
1	Resort Hotel	BB	Direct	Direct	C	No Deposit	Transient	2015	7	1
2	Resort Hotel	BB	Direct	Direct	A	No Deposit	Transient	2015	7	2
3	Resort Hotel	BB	Corporate	Corporate	A	No Deposit	Transient	2015	7	2
4	Resort Hotel	BB	Online TA	TA/TO	A	No Deposit	Transient	2015	7	3

```
# encoding categorical variables

cat_df['hotel'] = cat_df['hotel'].map({'Resort Hotel' : 0, 'City Hotel' : 1})

cat_df['meal'] = cat_df['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC': 3, 'Undefined': 4})

cat_df['market_segment'] = cat_df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/TO': 3,
                                                         'Complementary': 4, 'Groups': 5, 'Undefined': 6, 'Aviation': 7})

cat_df['distribution_channel'] = cat_df['distribution_channel'].map({'Direct': 0, 'Corporate': 1, 'TA/TO': 2, 'Undefined': 3,
                                                             'GDS': 4})

cat_df['reserved_room_type'] = cat_df['reserved_room_type'].map({'C': 0, 'A': 1, 'D': 2, 'E': 3, 'G': 4, 'F': 5, 'H': 6,
                                                         'L': 7, 'B': 8})

cat_df['deposit_type'] = cat_df['deposit_type'].map({'No Deposit': 0, 'Refundable': 1, 'Non Refund': 3})

cat_df['customer_type'] = cat_df['customer_type'].map({'Transient': 0, 'Contract': 1, 'Transient-Party': 2, 'Group': 3})

cat_df['year'] = cat_df['year'].map({2015: 0, 2014: 1, 2016: 2, 2017: 3})
```

```
# Display the categorical variables output
cat_df.head()
```

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day
0	0	0	0	0	0	0	0	0	7	1
1	0	0	0	0	0	0	0	0	7	1
2	0	0	0	0	1	0	0	0	7	2
3	0	0	1	1	1	0	0	0	7	2
4	0	0	2	2	1	0	0	0	7	3

```
# normalizing numerical variables

num_df['lead_time'] = np.log(num_df['lead_time'] + 1)
num_df['arrival_date_week_number'] = np.log(num_df['arrival_date_week_number'] + 1)
num_df['arrival_date_day_of_month'] = np.log(num_df['arrival_date_day_of_month'] + 1)
num_df['agent'] = np.log(num_df['agent'] + 1)
num_df['company'] = np.log(num_df['company'] + 1)
num_df['adr'] = np.log(num_df['adr'] + 1)
```

```
num_df.var()
```

```
lead_time                       2.582757
arrival_date_week_number        0.440884
arrival_date_day_of_month       0.506325
stays_in_weekend_nights         0.990258
stays_in_week_nights            3.599010
adults                          0.330838
children                        0.159070
babies                          0.009508
is_repeated_guest               0.030507
previous_cancellations          0.713887
previous_bookings_not_canceled  2.244415
agent                           3.535793
company                         1.346883
adr                             0.515480
required_car_parking_spaces     0.060201
total_of_special_requests       0.628652
dtype: float64
```

```
num_df['adr'] = num_df['adr'].fillna(value = num_df['adr'].mean())
```

```
X = pd.concat([cat_df, num_df], axis = 1)
y = df['is_canceled']
```

```
X.shape, y.shape
```

```
((119210, 26), (119210,))
```

```
# splitting data into training set and test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
```

```
X_train.head()
```

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	st
30408	0	0	5	0	1	0	2	2	11	21	4.753590	3.871201	2.890372	1	
84226	1	3	2	2	1	0	2	2	2	21	1.945910	2.197225	3.044522	0	
115183	1	0	2	2	1	0	0	3	7	3	5.135798	3.332205	1.098612	1	
62917	1	0	5	2	1	3	0	2	11	25	4.127134	1.609438	3.258097	0	
54644	1	0	0	0	5	0	0	2	1	18	5.267858	3.433987	3.135494	0	

```
X_test.head()
```

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	st
41128	1	0	3	2	1	0	0	0	8	11	0.693147	3.526361	2.564949	0	
112779	1	0	2	2	1	0	2	3	5	28	4.859812	3.091042	3.295837	0	
27244	0	0	1	1	1	0	0	2	8	24	1.609438	3.583519	3.178054	0	
98610	1	0	2	2	2	0	0	2	10	2	4.875197	3.713572	3.401197	0	
78196	1	0	0	0	1	0	0	0	10	6	2.564949	3.713572	1.386294	2	

```
# LR model
lr = LogisticRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

acc_lr = accuracy_score(y_test, y_pred_lr)
conf = confusion_matrix(y_test, y_pred_lr)
clf_report = classification_report(y_test, y_pred_lr)

print(f"Accuracy Score of Logistic Regression is : {acc_lr}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
Accuracy Score of Logistic Regression is : 0.8092721527836032
Confusion Matrix :
[[21263  1170]
 [ 5651  7679]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.95      0.86     22433
           1       0.87      0.58      0.69     13330

    accuracy                           0.81     35763
   macro avg       0.83      0.76      0.78     35763
weighted avg       0.82      0.81      0.80     35763
```

The performance of our LR model on the test data is as follows:
The Precision for class 1 (cancellations) is 87%, which means that approximately 87% of the bookings that the model predicted as canceled were actually canceled.
The Recall for class 1 is 58%, which means that the model correctly identified approximately 80% of the actual cancellations.
The F1-score for class 1 is 69%, which is the harmonic mean of Precision and Recall.

```
# KNN model
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)

acc_knn = accuracy_score(y_test, y_pred_knn)
conf = confusion_matrix(y_test, y_pred_knn)
clf_report = classification_report(y_test, y_pred_knn)

print(f"Accuracy Score of KNN is : {acc_knn}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
Accuracy Score of KNN is : 0.8903056231300506
Confusion Matrix :
[[21660   773]
 [ 3150 10180]]
Classification Report :
              precision    recall  f1-score   support

           0       0.87      0.97      0.92     22433
           1       0.93      0.76      0.84     13330

    accuracy                           0.89     35763
   macro avg       0.90      0.86      0.88     35763
weighted avg       0.89      0.89      0.89     35763
```

The performance of our KNN model on the test data is as follows:
The Precision for class 1 (cancellations) is 93%, which means that approximately 93% of the bookings that the model predicted as canceled were actually canceled.
The Recall for class 1 is 76%, which means that the model correctly identified approximately 76% of the actual cancellations.
The F1-score for class 1 is 84%, which is the harmonic mean of Precision and Recall.

```
# DT model
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

y_pred_dtc = dtc.predict(X_test)

acc_dtc = accuracy_score(y_test, y_pred_dtc)
conf = confusion_matrix(y_test, y_pred_dtc)
clf_report = classification_report(y_test, y_pred_dtc)

print(f"Accuracy Score of Decision Tree is : {acc_dtc}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
Accuracy Score of Decision Tree is : 0.946676732936275
Confusion Matrix :
[[21462   971]
 [  936 12394]]
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     22433
           1       0.93      0.93      0.93     13330

    accuracy                           0.95     35763
   macro avg       0.94      0.94      0.94     35763
weighted avg       0.95      0.95      0.95     35763
```

The performance of our DT model on the test data is as follows: The Precision for class 1 (cancellations) is 93%, which means that approximately 93% of the bookings that the model predicted as canceled were actually canceled. The Recall for class 1 is 93%, which means that the model correctly identified approximately 93% of the actual cancellations. The F1-score for class 1 is 93%, which is the harmonic mean of Precision and Recall.

The confusion matrix shows that there are still some False Positives and False Negatives, but the model is doing a relatively good job of minimizing them.

Additionally, the model is not overfitting, as the metric values for the test and train sets are close together, indicating that the model is generalizing well to unseen data.

```
# Random Forest model
rd_clf = RandomForestClassifier()
rd_clf.fit(X_train, y_train)

y_pred_rd_clf = rd_clf.predict(X_test)

acc_rd_clf = accuracy_score(y_test, y_pred_rd_clf)
conf = confusion_matrix(y_test, y_pred_rd_clf)
clf_report = classification_report(y_test, y_pred_rd_clf)

print(f"Accuracy Score of Random Forest is : {acc_rd_clf}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
ccuracy Score of Random Forest is : 0.9547017867628554
onfusion Matrix :
[22270   163]
[ 1457 11873]]
lassification Report :
              precision    recall  f1-score   support

           0       0.94      0.99      0.96     22433
           1       0.99      0.89      0.94     13330

    accuracy                           0.95     35763
   macro avg       0.96      0.94      0.95     35763
eighted avg       0.96      0.95      0.95     35763
```

The performance of our Random Forest (RF) model on the test data is as follows: The Precision for class 1 (cancellations) is 99%, which means that approximately 99% of the bookings that the model predicted as canceled were actually canceled. The Recall for class 1 is 89%, which means that the model correctly identified approximately 89% of the actual cancellations. The F1-score for class 1 is 94%, which is the harmonic mean of Precision and Recall.

The confusion matrix shows that there are some False Positives and False Negatives, but the model is doing a good job of minimizing them.

Additionally, the model is not overfitting, as the metric values for the test and train sets are close together, indicating that the model is generalizing well to unseen data.

```
# XGBoost Model Building
xgb = XGBClassifier(booster = 'gbtree', learning_rate = 0.1, max_depth = 5, n_estimators = 180)
xgb.fit(X_train, y_train)

y_pred_xgb = xgb.predict(X_test)

acc_xgb = accuracy_score(y_test, y_pred_xgb)
conf = confusion_matrix(y_test, y_pred_xgb)
clf_report = classification_report(y_test, y_pred_xgb)

print(f"Accuracy Score of XGBoost Classifier is : {acc_xgb}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
ccuracy Score of Ada Boost Classifier is : 0.9817409054050276
onfusion Matrix :
[[22419    14]
 [  639 12691]]
lassification Report :
              precision    recall  f1-score   support

           0       0.97      1.00      0.99     22433
           1       1.00      0.95      0.97     13330

    accuracy                           0.98     35763
   macro avg       0.99      0.98      0.98     35763
eighted avg       0.98      0.98      0.98     35763
```

The performance of our XGBoost (XGB) model on the test data is as follows:

The Precision for class 1 (cancellations) is 100%, which means that approximately 100% of the bookings that the model predicted as canceled were actually canceled. The Recall for class 1 is 95%, which means that the model correctly identified approximately 95% of the actual cancellations. The F1-score for class 1 is 97%, which is the harmonic mean of Precision and Recall.

The model is not overfitting, as the metric values for the test and train sets are close together, indicating that the model is generalizing well to unseen data. Compared to the Random Forest (RF) model, the XGBoost (XGB) model shows a slight improvement in all the performance metrics. Although the improvement is not significant, it is still better at balancing the trade-off between Precision and Recall, leading to a higher F1-score. This indicates that the XGBoost model is a better model for predicting cancellations.

```
y_train.head(), y_test.head()
```

```
(30408    0
 84226    0
 115183   0
 62917    1
 54644    1
 Name: is_canceled, dtype: int64,
 41128    1
 112779   0
 27244    0
 98610    0
 78196    0
 Name: is_canceled, dtype: int64)
```

```
# Compare the model output score
models = pd.DataFrame({
    'Model' : ['Logistic Regression', 'KNN', 'Decision Tree Classifier', 'Random Forest Classifier', 'XgBoost'],
    'Score' : [acc_lr, acc_knn, acc_dtc, acc_rd_clf,  acc_xgb]
})


models.sort_values(by = 'Score', ascending = False)
```

	Model	Score
4	XgBoost	0.981741
3	Random Forest Classifier	0.954702
2	Decision Tree Classifier	0.946677
1	KNN	0.890306
0	Logistic Regression	0.809272

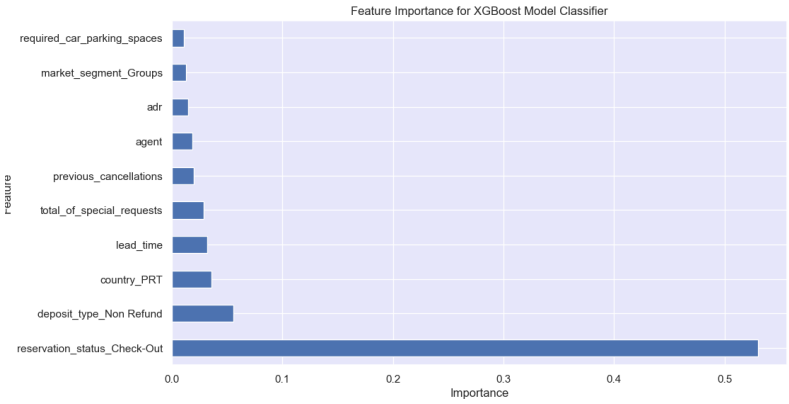
```
px.bar(data_frame = models, x = 'Score', y = 'Model', color = 'Score', template = 'plotly_dark', title = 'Models Comparison')
```

Among all the tested classifiers, XGBoost had the best performance in predicting hotel booking cancellations. Random forst classifier and Decision Tree Classifier also have good score but lesser compare to XGBoost.

```
# Feature importance for XGBoost Model classifier
xgb = XGBClassifier()
xgb.fit(X_train, y_train)

feature_importance = pd.Series(rf.feature_importances_, index=X.columns)
top_features = feature_importance.nlargest(10)  # Selecting top 10 features

# Plot feature importance
plt.figure(figsize=(10, 6))
top_features.plot(kind='barh')
plt.title('Feature Importance for XGBoost Model Classifier')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()
```



Interpretation of Results:

The accuracy score represents the proportion of correctly classified instances among all instances in the test set. Higher accuracy scores indicate better performance. Based on these results, the Ada Boost Classifier performs the best in terms of accuracy score, precision, and recall for both classes. It achieves the highest accuracy and the best balance between precision

and recall for both canceled and not canceled instances. These metrics suggest that the Logistic Regression model performs relatively well in predicting not canceled instances (class 0) but struggles with canceled instances (class 1), as indicated by the lower recall and F1-score for class 1.

Compared to the Random Forest (RF) model, the XGBoost (XGB) model shows a slight improvement in all the performance metrics. Although the improvement is not significant, it is still better at balancing the trade-off between Precision and Recall, leading to a higher F1-score. This indicates that the XGBoost model is a better model for predicting cancellations.

Overall, this model is performing well on the test data and seems to be a good model for predicting cancellations.

Initial Conclusion and Recommendation:

My target was to predict the cancellation of a hotel booking based of certain parameters from the hotel_bookings.csv dataset. The classification report generated from various models provides a summary of different evaluation metrics, including precision, recall, F1-score, and support, for each class. Precision (also called positive predictive value) is the proportion of true positive predictions among all positive predictions. It measures the accuracy of positive predictions and as we see the precision of XGBoost model is significantly high and that's the model I will use to predict the hotel booking cancellation. Recall is the proportion of true positive predictions among all actual positives. It measures the ability of the model to identify positive instances.

After training the XGBoost model, the feature importance is calculated. Feature importance represents the contribution of each feature to the model's predictions. Based on the plot, definitely reservation_status_check-out is the most dominating feature and has the most contribution in the model's predictions. Based on the outcome of this, I can definitely suggest XGBoost model is a good AI model which can predict the hotel booking cancellation.

DSC 630 Milestone 3: Data Selection and Project Proposal

Chitramoy Mukherjee

Predictive analysis on Hotel Booking Cancellation using

hotel_booking.csv dataset

Will I be able to answer the questions I want to answer with the data I have?

After reviewing columns and rows in the dataset to ensure it contains relevant information such as booking details, guest demographics, room types, reservation status booking dates, guest characteristics, and booking outcomes will be able to create the predictive model to answer the questions we are looking to answer for this project. By carefully evaluating Data Content, Question Suitability, Data Quality and Predictive Power factors we can determine "hotel_bookings.csv" dataset can support your predictive analysis and help answer your desired questions about hotel bookings. If the dataset lacks certain information or questions require additional data, we may need to explore other sources or adjust your analysis approach accordingly.

What visualizations are especially useful for explaining my data?

Visualizations play a crucial role in exploring and explaining the patterns and insights within hotel booking dataset. Below visualizations you can create to understand and communicate your data effectively:

1. **Bar Charts**:

- Visualize categorical variables such as hotel type (city or resort), customer type (transient, contract, group), and room type.

- Compare counts or percentages across different categories to identify trends and preferences.

2. **Histograms**:
- Explore the distribution of numerical variables such as lead time (number of days between booking and arrival), stays on weekend nights, stays on week nights, etc.
- Understand the frequency and spread of values within each variable.

3. **Pie Charts**:
- Show the distribution of categorical variables like customer types, market segments, or distribution channels. Highlight the proportion of each category within the dataset.

4. **Box Plots**:
- Visualize the distribution of numerical variables such as lead time, booking duration, or room rate, across different categories like hotel type or customer type. Identify outliers, quartiles, and overall distribution characteristics.

5. **Time Series Plots**:
- Analyze temporal patterns by plotting variables like booking date, arrival date, or lead time over time. Identify seasonal trends, booking peaks, or patterns in cancellations.

6. **Scatter Plots**:
- Explore relationships between numerical variables, such as lead time and booking duration, or between numerical and categorical variables. Identify correlations or patterns in the data.

7. **Heatmaps**:
- Visualize correlations between variables in the dataset, especially useful for understanding relationships between numerical variables. Identify areas of high or low correlation, which can inform feature selection for predictive modeling.

Do I need to adjust the data and/or driving questions?

Adjusting the data and refining driving questions may be necessary to ensure the success and relevance of your predictive analysis. Need to perform some basic adjustments in the dataset to Check for missing values, duplicates, and inconsistencies in the dataset. Clean the data by filling missing values, removing duplicates, and resolving inconsistencies to ensure data quality. Create new features from existing ones if they can provide additional predictive power. I can derive features like total stays (weekend nights + weeknights), booking lead time, or booking season from the available data. Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding. This allows to include categorical variables in predictive models effectively.

Do I need to adjust my model/evaluation choices?

Will Consider different types of predictive models suitable for the dataset and objectives. Common models for hotel booking prediction include logistic regression, decision trees, random forests, gradient boosting machines (GBM), support vector machines (SVM), and neural networks. Balance between model complexity and interpretability based on our needs and priorities. While complex models like neural networks may offer high predictive performance, simpler models like logistic regression or decision trees might be easier to interpret and explain. Will explore ensemble methods like random forests or gradient boosting to combine predictions from multiple models. Ensemble methods often improve predictive performance and robustness compared to individual models. Will evaluate metrics that align with the goals of the predictive analysis. Common metrics for binary classification tasks in hotel booking prediction include accuracy, precision, recall, F1 score, ROC-AUC, and precision-recall curve. Will Evaluate model performance in the context of business relevance. Consider how well the predictive model aligns with the ultimate business goals and whether the predictions are actionable and useful for decision-making.

Are my original expectations still reasonable?

Our initial expectation is to build a predictive model using the hotel_booking.csv data to determine whether a hotel booking would be canceled, which is crucial for hotels as cancellations affect revenue and operational planning. After reviewing the dataset and attributes and volume of data available to build the model, expectation seems to be reasonable. Will be able to answer the questions from the available data and attributes. If I need some reference data along with this data to answer my questions, I will source it during my EDA and model development part.

DSC 630 Milestone 2: Data Selection and Project Proposal

Chitramoy Mukherjee

Predictive analysis on Hotel Booking Cancellation using hotel_booking.csv dataset

Predictive analysis in hotel booking cancellation involves using historical data and statistical techniques to forecast the likelihood of a guest canceling their reservation before their scheduled arrival date. In this project, we aim to build a predictive model to determine whether a hotel booking would be canceled, which is crucial for hotels as cancellations affect revenue and operational planning. Dataset contains many features related to booking, such as lead time, deposit type, and special requests, which adds to the model's complexity. This prediction model will help the business to determine the probability of cancellation of a booking and create a backup plan to overcome any loss. From a customer perspective we can analyze what will be the best time to book a hotel and get the best pricing for it.

Dataset information and steps:

Hotel_booking.csv dataset contains 2015-2017 timeframe data for City hotel and Resort hotel booking. Below are the variables and its description from hotel_booking.csv dataset.

Index	Variable	Description
1	**hotel**	Type of hotel (Resort Hotel, City Hotel)
2	**is_canceled**	Reservation cancellation status (0 = not canceled, 1 = canceled)
3	**lead_time**	Number of days between booking and arrival
4	**arrival_date_year**	Year of arrival
5	**arrival_date_month**	Month of arrival
6	**arrival_date_week_number**	Week number of the year for arrival
7	**arrival_date_day_of_month**	Day of the month of arrival
8	**stays_in_weekend_nights**	Number of weekend nights (Saturday and Sunday) the guest stayed or booked
9	**stays_in_week_nights**	Number of weeknights the guest stayed or booked
10	**adults**	Number of adults
11	**children**	Number of children
12	**babies**	Number of babies
13	**meal**	Type of meal booked (BB, FB, HB, SC, Undefined)
14	**country**	Country of origin of the guest
15	**market_segment**	Market segment designation
16	**distribution_channel**	Booking distribution channel
17	**is_repeated_guest**	If the guest is a repeat customer (0 = not repeated, 1 = repeated)
18	**previous_cancellations**	Number of previous bookings that were canceled by the customer
19	**previous_bookings_not_canceled**	Number of previous bookings that were not canceled by the customer
20	**reserved_room_type**	Type of reserved room
21	**assigned_room_type**	Type of assigned room
22	**booking_changes**	Number of changes made to the booking
23	**deposit_type**	Type of deposit made (No Deposit, Refundable, Non Refund)
24	**agent**	ID of the travel agent responsible for the booking
25	**company**	ID of the company responsible for the booking
26	**days_in_waiting_list**	Number of days the booking was in the waiting list
27	**customer_type**	Type of customer (Transient, Contract, Transient-Party, Group)
28	**adr**	Average Daily Rate
29	**required_car_parking_spaces**	Number of car parking spaces required
30	**total_of_special_requests**	Number of special requests made

31	**reservation_status**	Last reservation status (Check-Out, Canceled, No-Show)
32	**reservation_status_date**	Date of the last reservation status
33	**name**	Guest's name
34	**email**	Guest's email address
35	**phone-number**	Guest's phone number
36	**credit_card**	Last four digits of the guest's credit card

Below are the key steps involved for project execution,

Historical Data: Historical booking data serves as the foundation for predictive analysis. It includes information on past reservations, cancellations, guest demographics, booking channels, and other relevant variables. Data needs to be cleaned up and handled and loaded for further processing.

Feature Selection: Identifying relevant features or variables that influence cancellation behavior is crucial. These may include lead time, booking channel, seasonality, room type, price, guest demographics, and external factors such as events or holidays.

Model Development: Predictive models, such as logistic regression, decision trees, random forests, or neural networks, are trained on historical data to predict the likelihood of cancellation for future bookings. These models learn from patterns and relationships in the data to make predictions.

Model Evaluation: Models are evaluated using metrics such as accuracy, precision, recall, or area under the ROC curve (AUC) to assess their predictive performance. Cross-validation techniques help validate the model's generalizability.

Implementation and Deployment: Once a predictive model is developed and validated, it can be deployed into hotel management systems to generate real-time predictions for upcoming bookings. These predictions inform decision-making processes related to pricing, inventory management, and resource allocation.

Significance:

Predictive analysis helps hotels optimize pricing strategies by adjusting room rates dynamically based on predicted cancellation probabilities and demand fluctuations. Anticipating cancellations allows hotels to better manage inventory, allocate resources efficiently, and minimize the impact on operations. By proactively managing cancellations, hotels can minimize overbooking situations and ensure a smoother guest experience, ultimately leading to higher guest satisfaction and loyalty. Predictive insights enable hotels to make informed decisions regarding marketing campaigns, promotions, and capacity planning, thereby maximizing revenue potential and competitiveness in the market.

Types of Models:

For this project, I plan to utilize several machine learning models to predict the chances of cancellation of hotel booking. Will implement and tune classification models including Decision Trees, Random Forest, and XGBoost. Will Emphasize achieving high F1-score for class 1, ensuring comprehensive identification of booking cancellations.

Evaluation of Results:

Will select evaluation metrics suitable for the binary classification problem of cancellation prediction. Common metrics includes below,

- Accuracy: Measures overall correctness of predictions.
- Precision: Indicates the proportion of predicted cancellations.
- Recall: Measures the proportion of actual cancellations that are correctly predicted.
- F1 Score: Harmonic means of precision and recall, useful for imbalanced datasets.

- Area Under the Receiver Operating Characteristic Curve (AUC-ROC): Measures the model's ability to discriminate between positive and negative classes.

Learning Objectives:

Through this project, I aim to gain a deeper understanding of predictive analytics techniques and their application Identify trends and factors influencing booking cancellations to anticipate future cancellations more accurately. Gain insights into optimizing revenue management and resource allocation strategies based on predicted cancellation probabilities. Understand guest behavior and preferences regarding cancellations to enhance guest satisfaction and improve overall hotel operations.

Risks and Ethical Implications:

One of the major risks for utilizing guest data for predictive analysis raises privacy risks, necessitating strict adherence to data protection regulations and ensuring secure handling of sensitive information to prevent unauthorized access or misuse. There's a risk of algorithmic bias leading to unfair treatment of certain groups, such as discriminating against guests from specific demographics or regions. Ethical considerations demand the identification and mitigation of biases to ensure fair and equitable predictions. Lack of transparency in the predictive modeling process can erode trust and lead to unintended consequences. Ensuring transparency, accountability, and clear communication about the model's limitations and implications is essential to maintain ethical standards and stakeholder trust.

Contingency Plan:

If the original project plan does not work out as expected, I will reassess the data sources and modeling techniques to identify alternative approaches. This may involve exploring additional datasets, adjusting preprocessing steps, or experimenting with different machine learning

algorithms. Regular communication with project advisors and peers will help in troubleshooting and adapting to any challenges encountered during the project. If I must change the topic if the results are not expected, I will work on the breast cancer prediction modeling. I have checked the dataset on that which I can use for this project.

Additional Considerations:

In addition to developing predictive models, I plan to explore feature importance to understand which variables contribute most to readmission risk. Furthermore, I will visualize model predictions and insights to facilitate interpretation and decision-making by healthcare professionals. Continuous refinement of the predictive models based on real-world feedback and new data will be essential for improving their accuracy and applicability in clinical settings.

References:

Dataset hotel_booking.csv is sourced from https://www.kaggle.com.