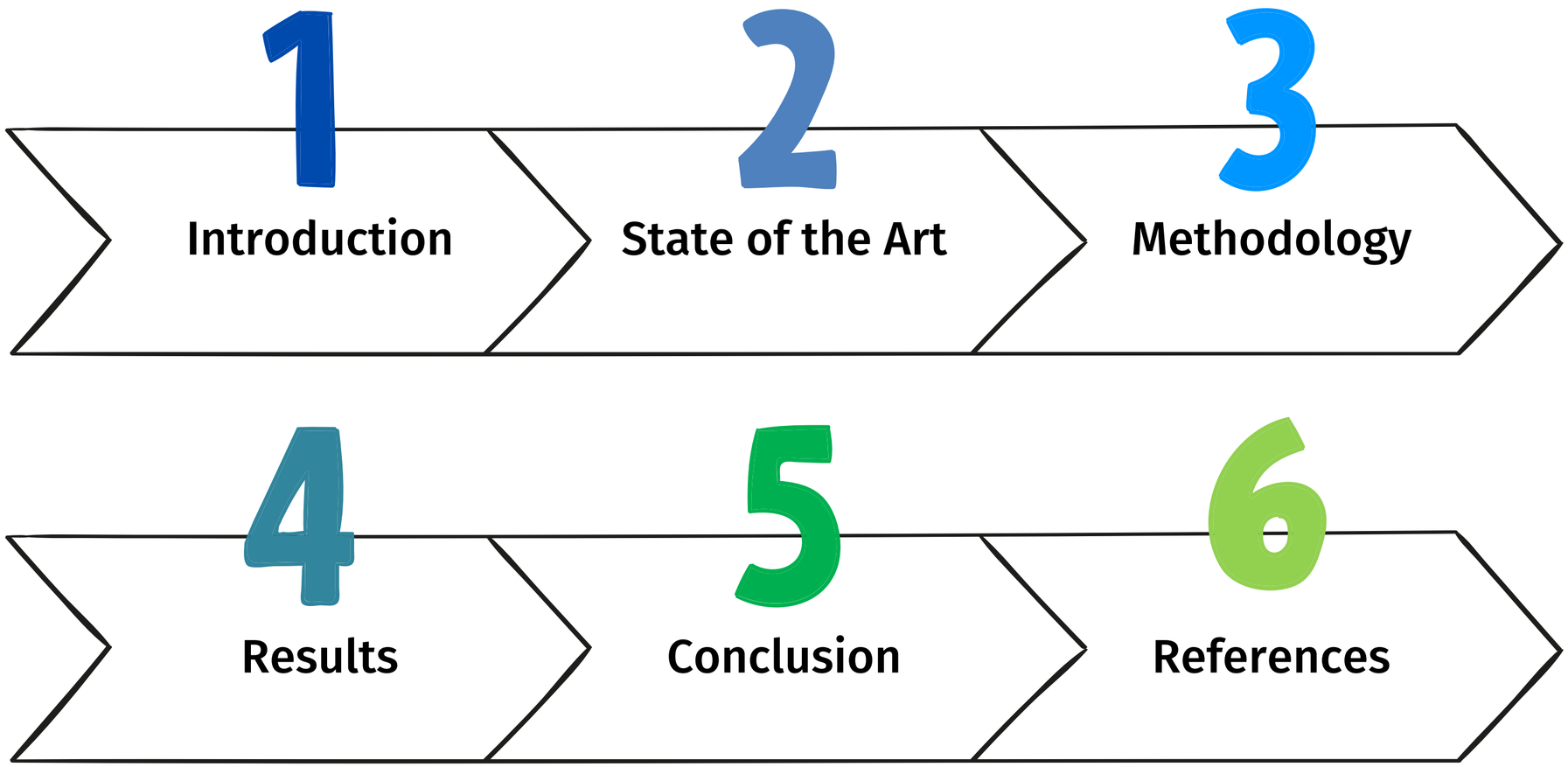
CAN WE PREDICT HOTEL BOOKING CANCELLATIONS ?

Analysis & Prediction of Hotel Booking Cancellation With Ensemble Learning

Presented By

Name : H.H.J.Karunarathna
Index No : 18/ENG/050

Table of Content



01.Introduction

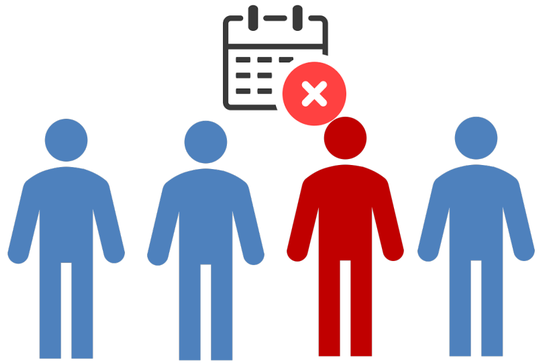
Problem

Hotel industry is one of the faster growing businesses of tourism sector.

In **2019**, worth over

$570B

Source : [1]



1 out of **4** guests are

Cancelling

Source : [2]

Rising rate of cancellation have a negative impact on the hotel industry.

- Loss of income in shape of unsold room
- Lower RevPAR
- Operational problems

⬇$8.7B/Year

From cancellations & no-shows

Source : [3]

To mitigate the impact of cancellations,
hotels must identify potentially cancelled bookings.

Objectives

1. Build an ensemble learning-based classification model that predicts hotel booking cancellations with the highest possible accuracy.

2. Conduct an EDA to identify patterns in hotel booking cancellations.

3. Develop a REST API to expose the model's functionality through an HTTP endpoint.

02.State of Art

Research Works	Data Used	Models Built	Addressing Class Imbalance	Ensemble Learning
Predicting Hotel Booking Cancellation To Decrease Uncertainty And Increase Revenue(2017) - Antonio Et.Al [4]	Real booking data from 4 hotels	Boosted Decision Trees, Decision Forest, Decision Jungle, Locally Deep Support Vector Machine, Neural Networks	✗	✗
An Automated Machine Learning Based Decision Support System to Predict Hotel Booking Cancellation (2019) - Antonio Et.Al [5]	All reservations on-the-books	XGBoost	✗	✗
Performance Analysis of Machine Learning Techniques to Predict Hotel Booking Cancellations in Hospitality Industry (2020) - Shagriarr Et.Al [6]	Real booking data from 2 hotels	XGBoost, Decision Tree, Gradient Boost, Random forest, Linear Regression, KNN	✗	✗
Solving the Problem of Class Imbalance in the Prediction of Hotel Cancellations (2021- Mohtd Et.Al [7]	Real booking data from 2 hotels	Linear Regression, Decision Tree, AdaBoost, Gradient Boost, Random forest	✓	✗

03.Methodology

01.Data Selection

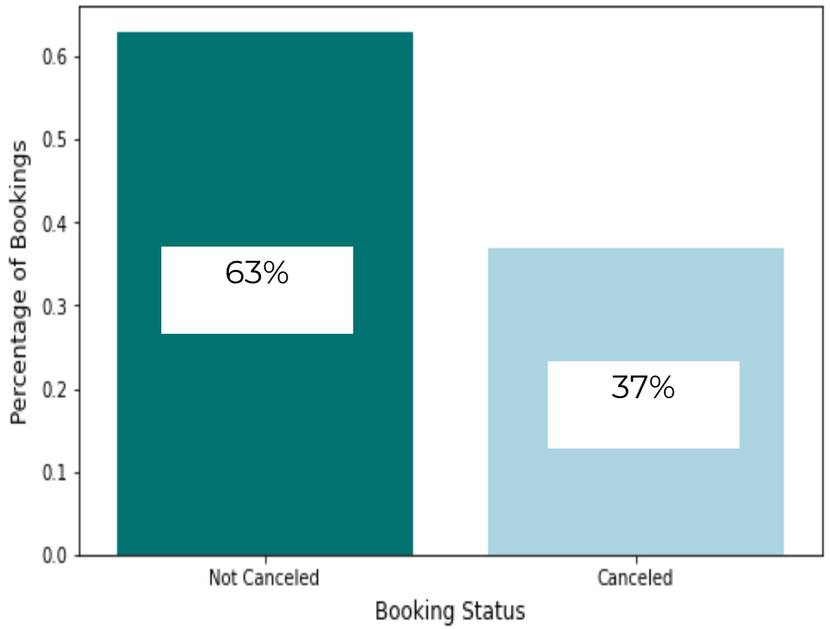
Dataset : Open hotel booking demand dataset

Created by Antonio, Almeida and Nunes ,2019

Records related to both a Resort Hotel & City Hotel

Size : 32 columns & ~120K observations

Imbalance : Slightly imbalanced




```
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
```

02. Data Cleaning

01. Handling Missing Values

	missing_percentage %
children	0.003350
country	0.408744
agent	13.686238
company	94.306893

```
# Changing agent value of "NULL" to "No Agent":
df['agent'].fillna("No Agent", inplace=True)
```

✓ 0.0s

```
# Changing company value of "NULL" to "No Company":
df['company'].fillna("No Company", inplace=True)
```

✓ 0.0s

```
# Drop the 4 observations with missing children values.
df.dropna(subset=['children'], inplace=True)
```

✓ 0.1s

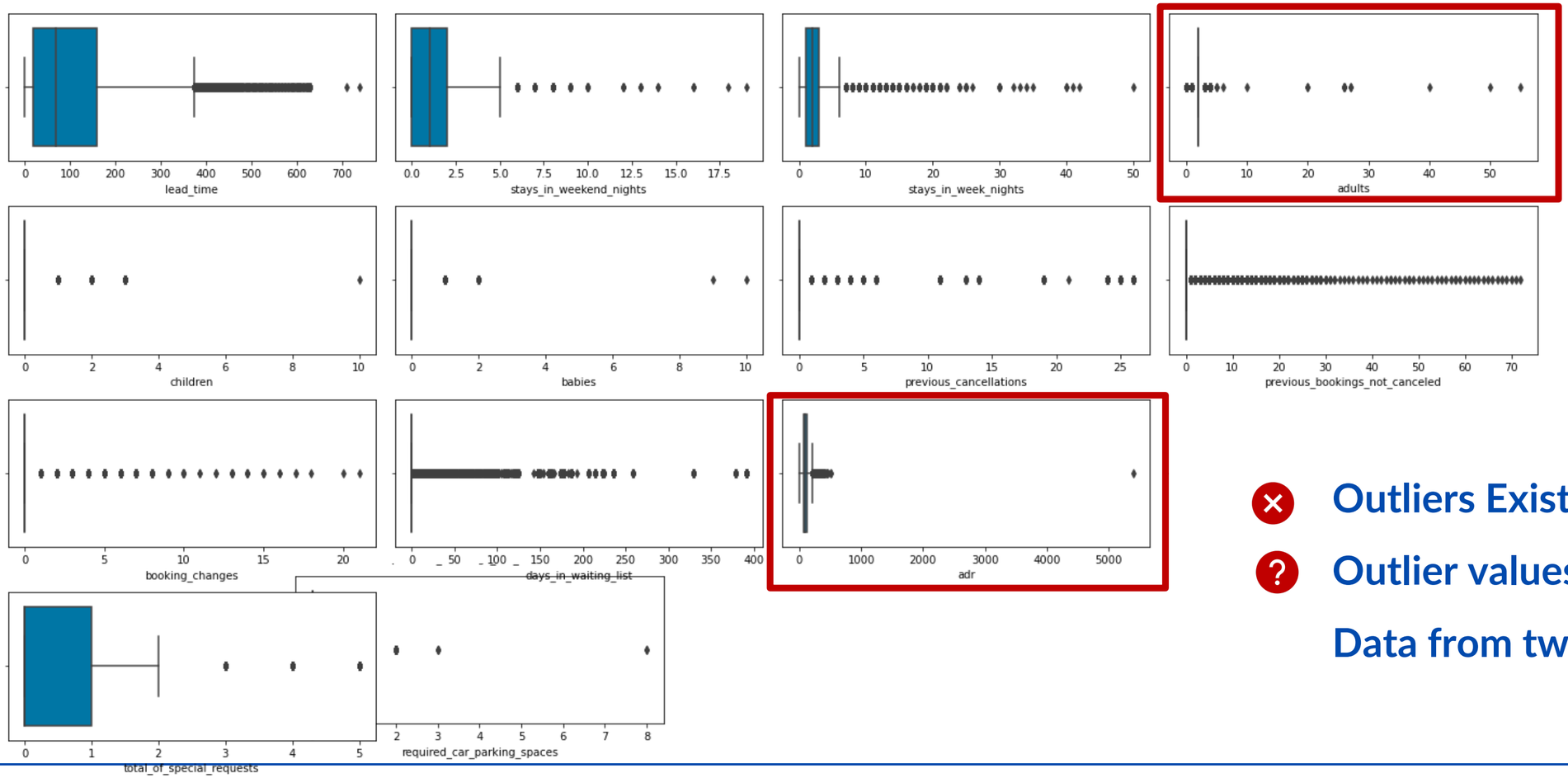
	missing_percentage %
children	0.0
country	0.0
agent	0.0
company	0.0

```
# Changing country value of "NULL" to "Unknown":
df['country'].fillna("Unknown", inplace=True)
```

✓ 0.0s

02. Data Cleaning

02. Removing Outliers

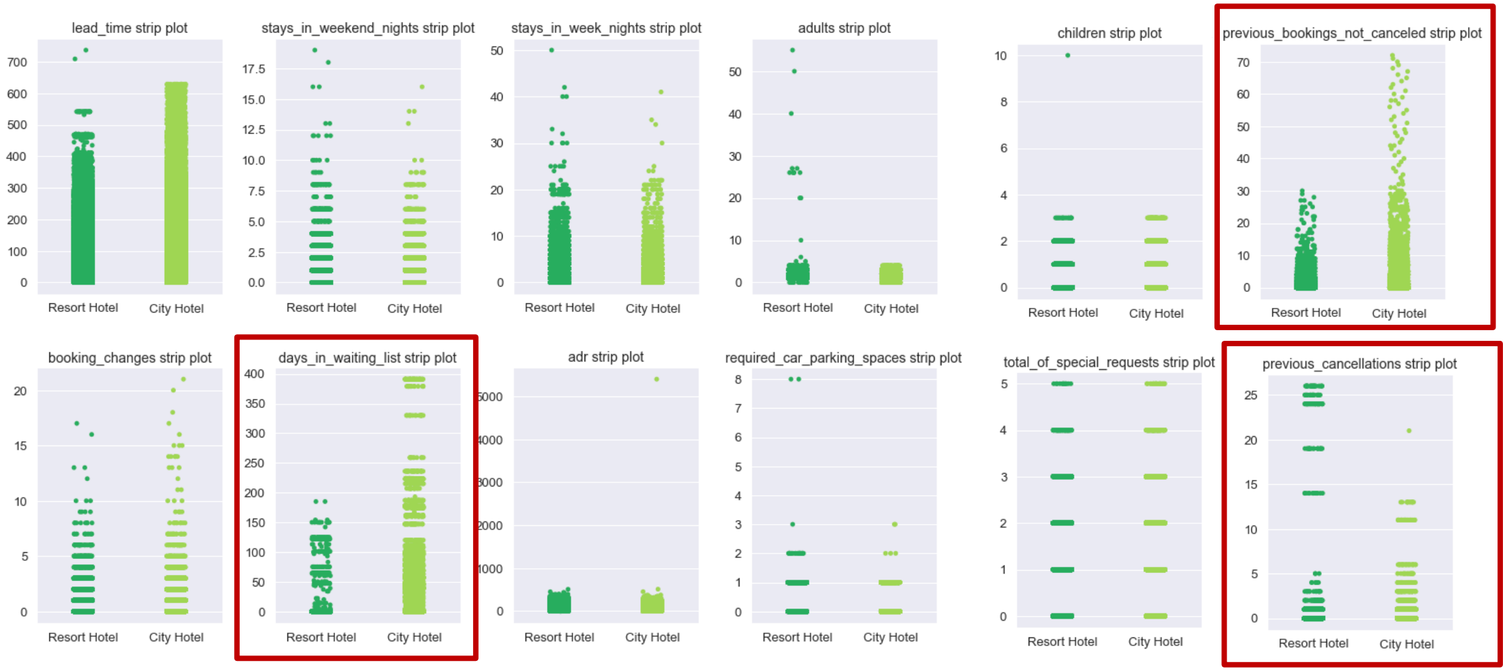


- Outliers Exists
- Outlier values are quite similar

  Data from two hotels ??

02. Data Cleaning

Creating Strip-plot to visualize differences in data distribution between the 2 hotels



- Dropped rows with adr below 0 and above 5000
- Dropped rows with 0 adults
- Converted columns with unusual values into categorical
  - days_in_waiting_list,
  - previous_cancellations,
  - previous_bookings_not_canceled

02. Data Cleaning

03. Removing Inconsistencies

```
df['meal'].unique()
```

✓ 0.0s

```
array(['BB', 'FB', 'HB', 'SC', 'Undefined'], dtype=object)
```

stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment
2	7	2	0.0	0	BB	GBR	Offline TA/TO
0	2	1	0.0	0	BB	PRT	Offline TA/TO
0	1	1	0.0	0	BB	PRT	Direct
1	5	2	0.0	0	BB	ESP	Online TA
1	0	1	0.0	0	BB	ESP	Online TA
2	2	2	0.0	0	BB	CN	Offline TA/TO
0	3	2	0.0	0	Undefined	PRT	Groups
0	3	2	0.0	0	SC	GBR	Offline TA/TO
0	3	1	0.0	0	BB	FRA	Corporate
1	3	2	0.0	0	BB	PRT	Offline TA/TO
2	1	2	0.0	0	BB	PRT	Offline TA/TO

```
#Replace Undefined with SC- self catering
df['meal'] = df['meal'].str.replace('Undefined', 'SC')
df['meal'].unique()
```

✓ 0.1s

```
array(['BB', 'FB', 'HB', 'SC'], dtype=object)
```

03. Data Transformation

01. Feature Engineering

stays_in_weekend_nights	stays_in_week_nights
1	1
0	1
1	2
0	1
1	4

total_stays
2
1
3
1
5

- Combined Features
- Dropped redundant coloumns

adults	children	babies
1	0.0	0
3	1.0	0
3	0.0	0
2	0.0	0
2	0.0	0

guests
1.0
4.0
3.0
2.0
2.0

children	babies
0.0	0
0.0	0
0.0	0
1.0	1
0.0	0

kids
0.0
0.0
0.0
2.0
0.0

03. Data Transformation

02. Feature Selection

```
hotel : 2 labels
arrival_date_month : 12 labels
meal : 4 labels
country : 178 labels
market_segment : 7 labels
distribution_channel : 5 labels
reserved_room_type : 9 labels
assigned_room_type : 11 labels
deposit_type : 3 labels
agent : 334 labels
company : 349 labels
customer_type : 4 labels
reservation_status : 3 labels
reservation_status_date : 926 labels
```

14 Categorical Features

1

	agent	company
2391	240.0	113.0
5694	405.0	405.0
8756	223.0	223.0
12230	5.0	250.0
14083	196.0	61.0
14376	240.0	268.0
14391	240.0	331.0
16637	240.0	268.0
17500	184.0	146.0
18620	250.0	399.0

```
df=df.drop(columns=["agent","company"])
```

2

reserved_room_type	assigned_room_type
C	C
C	C
A	C
A	A
A	A

```
df=df.drop(columns=["reserved_room_type","assigned_room_type"])
```

3

```
df=df.drop(columns=["country"])
```

4

```
df=df.drop(columns=["reservation_status_date"])
```

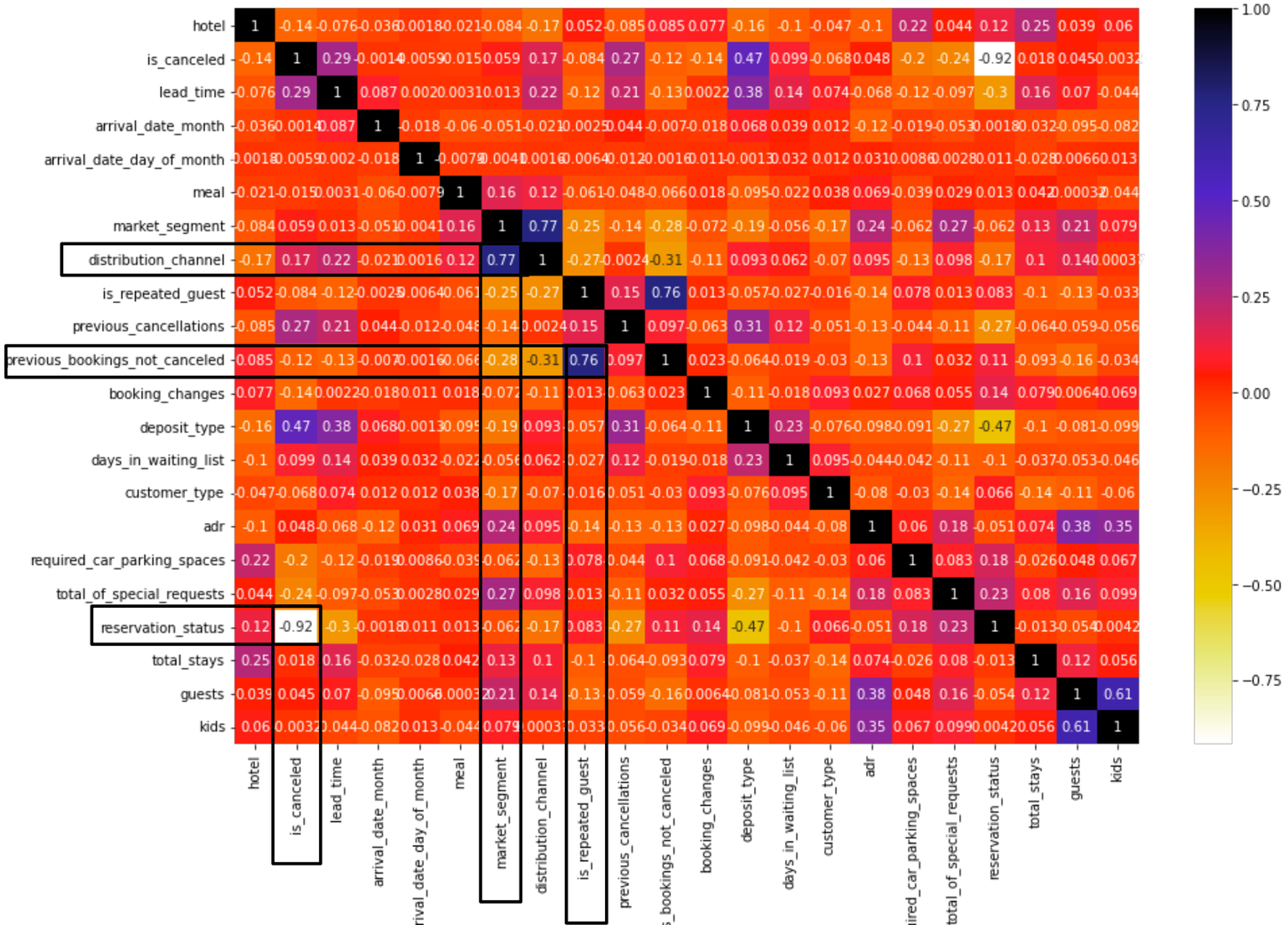
5

```
df=df.drop(columns=["arrival_date_year"])
```

- The agent and company IDs are recorded
- No information about Room Types
- “country" feature excluded to avoid bias towards cancellation rates of specific countries.
- Outdated features are removed

03. Data Transformation

03. Removing Correlated Features



Highly Correlated !

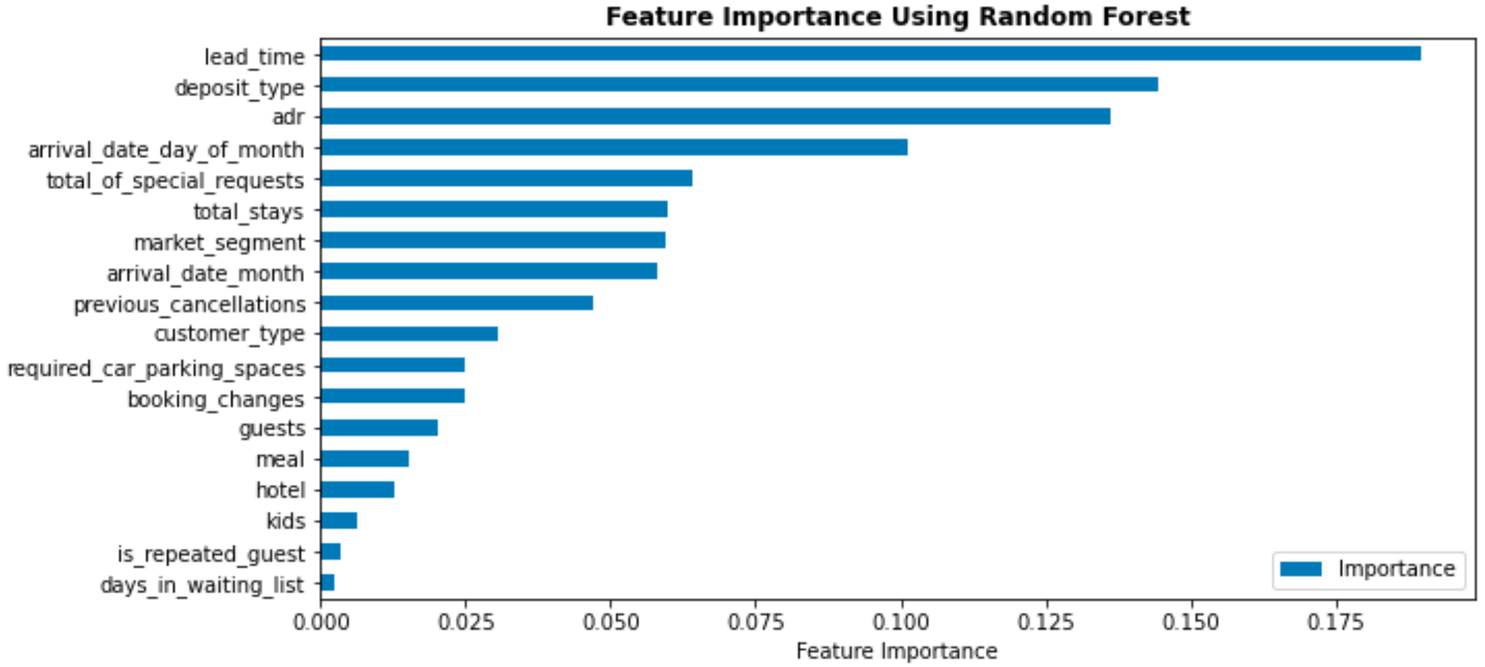
1. distribution_channel ✕
   market_segment

2. previous_bookings_not_cancelled ✕
   is_repeated_guest

3. reservation_status ✕
   is_cancelled

03. Data Transformation

04. Feature Importance




```
Feature ranking:
1. lead_time
2. deposit_type
3. adr
4. arrival_date_day_of_month
5. total_of_special_requests
6. total_stays
7. market_segment
8. arrival_date_month
9. previous_cancellations
10. customer_type
11. required_car_parking_spaces
12. booking_changes
```

To 12 features for the model

03. Data Transformation

05. One Hot Encoding & Creating Dummy Variables

Arrival Month

```
dummies = pd.get_dummies(df.arrival_date_month,prefix='arrival_date_month')
df = pd.concat([df,dummies],axis='columns')
df = df.drop('arrival_date_month',axis='columns')
```

arrival_date_month_February	...	arrival_date_month_January	arrival_date_month_July
0	...	0	1
0	...	0	1
0	...	0	1
0	...	0	1
0	...	0	1

Deposit Type

```
dummies = pd.get_dummies(df.deposit_type,prefix='deposit_type')
df = pd.concat([df,dummies],axis='columns')
df = df.drop('deposit_type',axis='columns')
```

deposit_type_No Deposit	deposit_type_Non Refund	deposit_type_Refundable
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0

Market Segment

```
dummies = pd.get_dummies(df.market_segment,prefix='market_segment')
df = pd.concat([df,dummies],axis='columns')
df = df.drop('market_segment',axis='columns')
```

market_segment_Aviation	market_segment_Complementary	market_segment_Corporate	market_segment_Direct
0	0	0	1
0	0	0	1
0	0	0	1
0	0	1	0
0	0	0	0

Customer Type

```
dummies = pd.get_dummies(df.customer_type,prefix='customer_type')
df = pd.concat([df,dummies],axis='columns')
df = df.drop('customer_type',axis='columns')
```

customer_type_Contract	customer_type_Group	customer_type_Transient
0	0	1
0	0	1
0	0	1
0	0	1
0	0	1

04. Model Building

01. Selection of Evaluation Metric

- On average each room costs ~$102
- Median number of nights stayed by a guest is 3.
- Gross profit =$306

Assuming a revenue of 40% from gross profit, the hotel can expect to make $123/stay

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

	True Observation	False Observation
True Prediction	$123.0 (Predict canceled, cancelled) TP	-$183.00 (Predict canceled, not cancelled) FP
False Prediction	-$1830.00 (Predict not canceled, cancelled) FN	$123.0 (Predict not canceled, not cancelled) TN

Risk of Flase Postives (-$183) and False Negatives (-$1830)

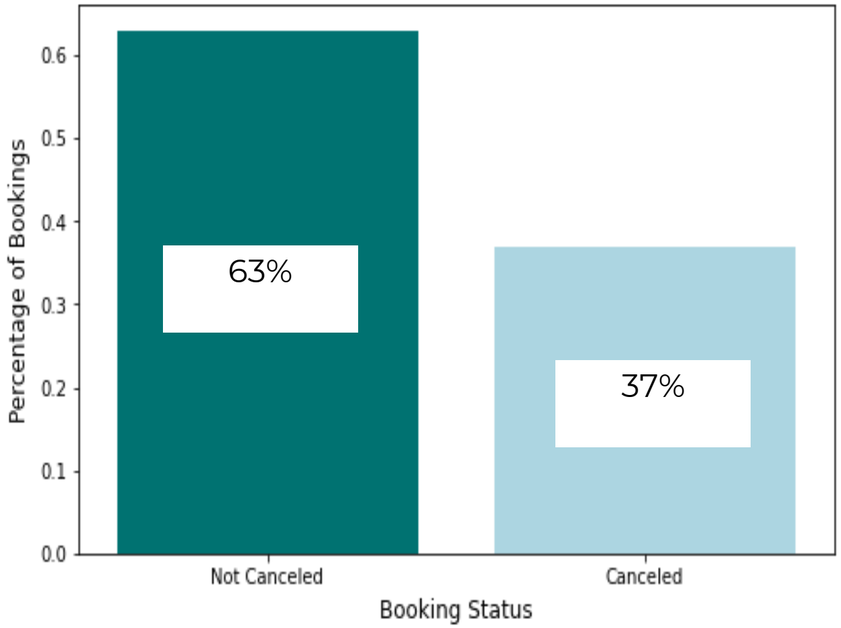
Balancing false positive and false is a necessity

Accuracy ✓

F1-Score ✓

04. Model Building

02. Data Imbalance Analysis



Imbalance in the classes is **not extreme**, but it is not insignificant.

Compare several different types of balancing in combination with several different types of machine learning models

Balancing Methods

- None
- Random Under Sampling
- Random Oversampling
- SMOTE
- SMOTE + Random Under Sampling

Machine Learning Models

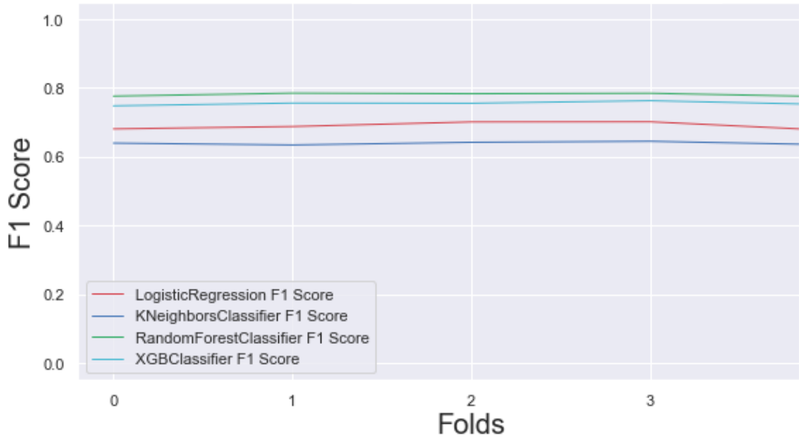
- **Logistic Regression** as basic binary classification model
- **KNN** as instance-based machine learning model
- **Random Forest** as bagging ensemble method
- **XGBoost** as boosting ensemble method

Which Balancing Method Works ??

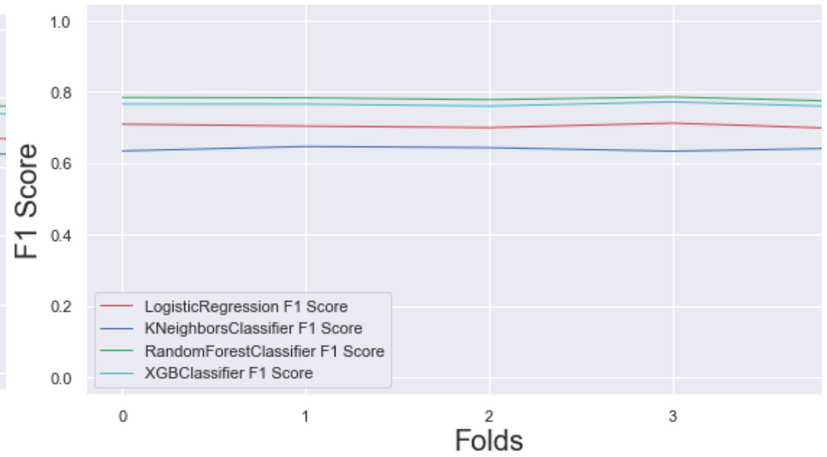
04. Model Building

Which Balancing Method Works ?

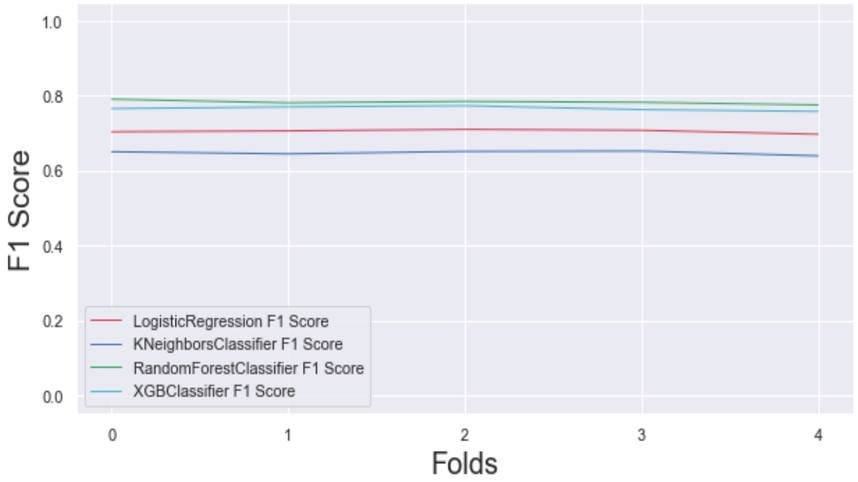
No Balancing



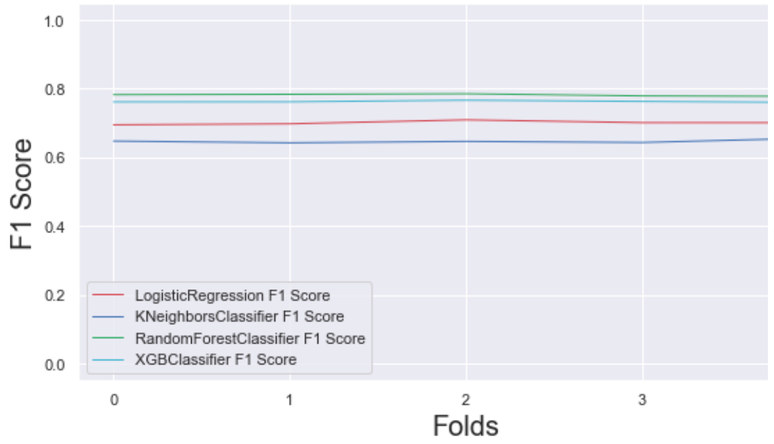
Undersampling



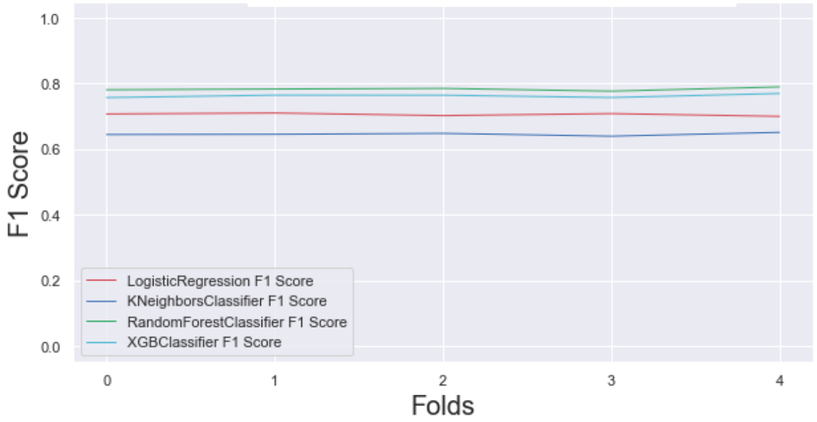
Oversampling



SMOTE



SMOTE + Undersampling



Not Balancing ✓

04. Model Building

03. Machine Learning Models

1. Logistic Regression
2. Support Vector Classifier (SVC)
3. Decision Tree
4. Random Forest
5. eXtreme Gradient Boosting (XGB)
6. Light Gradient Boosting (LGB)
7. Bagging Classifier

	training	testing
Logistic Regresion	0.798021	0.799574
SVC	0.687823	0.688911
Decision Tree	0.992340	0.794251
Random Forest	0.992340	0.847705
XGB	0.865187	0.836947
LGB	0.843455	0.833978
Bagging Classifier	0.979684	0.836387

Overfitting

Evaluation Matrix

	Logistic Regression	SVC	Decision Tree	Random Forest	XGB	LGB	Bagging Classifier
Accuracy	0.799574	0.688911	0.794251	0.847705	0.836947	0.833978	0.836387
Recall	0.571755	0.252777	0.726982	0.721656	0.682240	0.658499	0.698980
Precision	0.831195	0.721233	0.717806	0.842124	0.845052	0.857511	0.829810
F1 Score	0.677486	0.374352	0.722365	0.777250	0.754968	0.744943	0.758797

04. Model Building

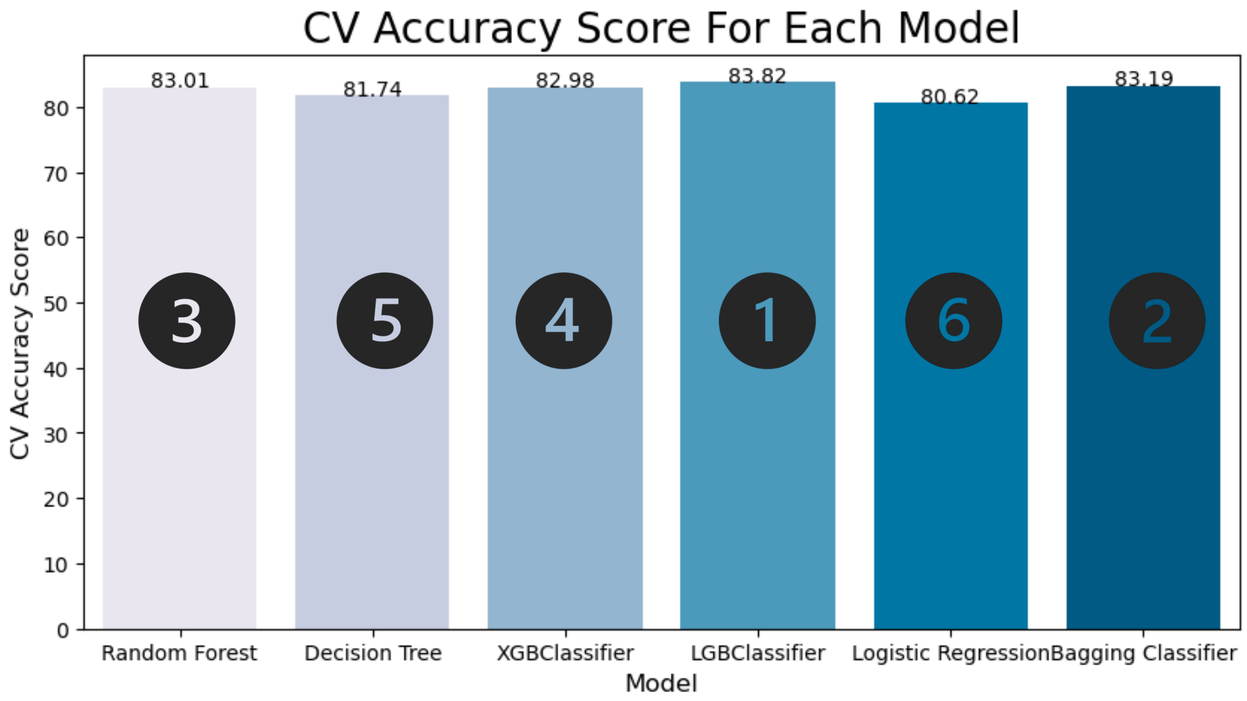
04. Hyper Parameter Tuning with Grid Search CV & Results

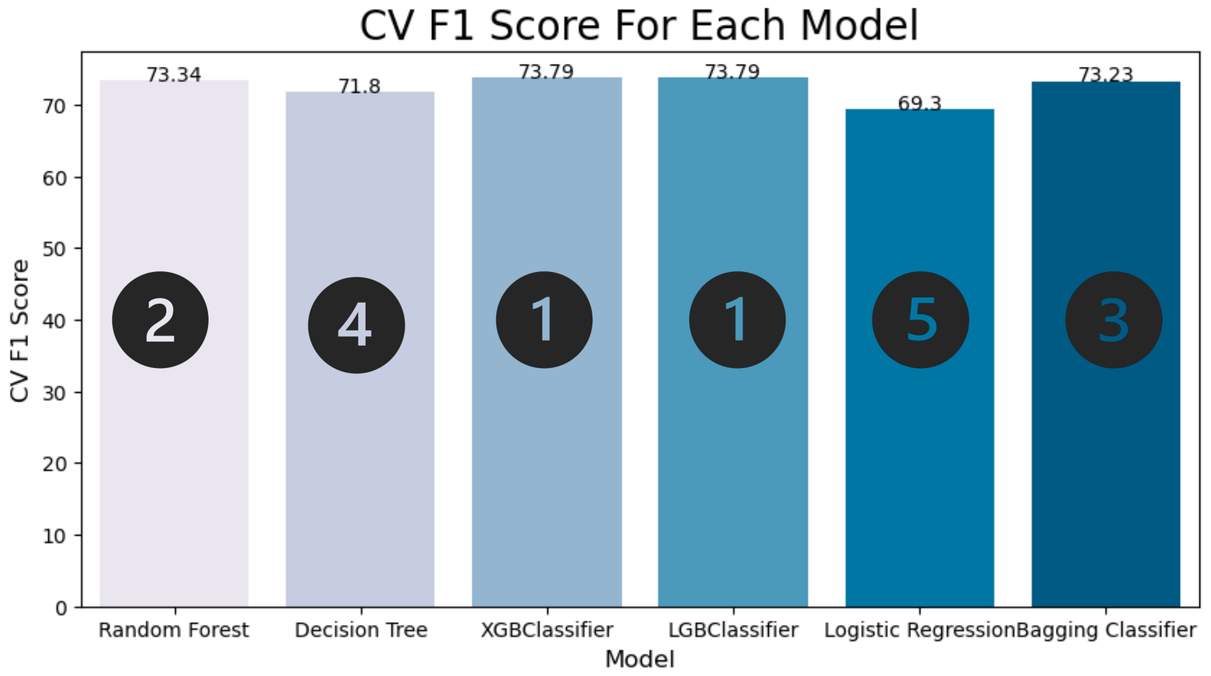
Tuned Models

	Best Params	Training Accuracy	Testing Accuracy	F1
Logistic Regression	{'C': 2.0, 'class_weight': None, 'max_iter': 2000, 'penalty': 'l2'}	0.808827	0.807475	0.696681
Decision Tree	{'max_depth': 15, 'min_samples_leaf': 12, 'min_samples_split': 20}	0.838412	0.819129	0.728694
Random Forest	{'max_depth': 15, 'min_samples_leaf': 2, 'min_samples_split': 4}	0.835675	0.827926	0.732281
XGB	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}	0.837332	0.829327	0.740235
LGB	{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 500}	0.864179	0.835715	0.756640
Bagging Classifier	{'max_features': 0.5, 'max_samples': 0.2, 'n_estimators': 50}	0.824556	0.819858	0.718304

04. Model Building

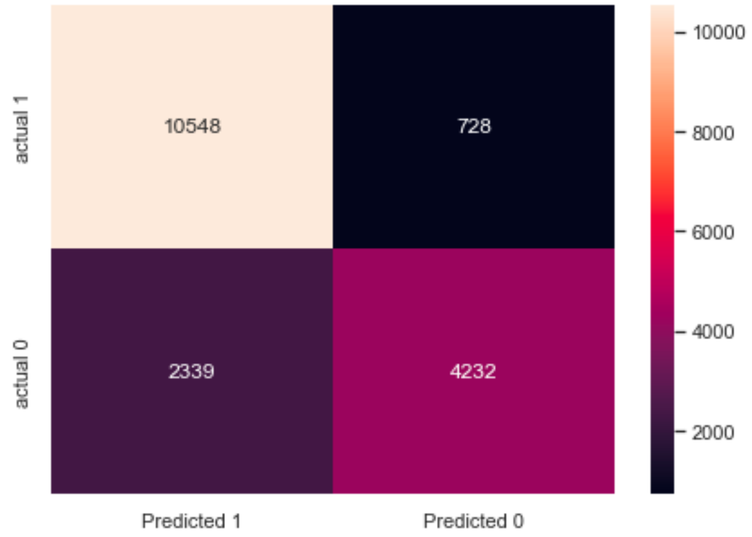
05. Dataframes to represent Cross Validation Score of Accuracy and F1-Score





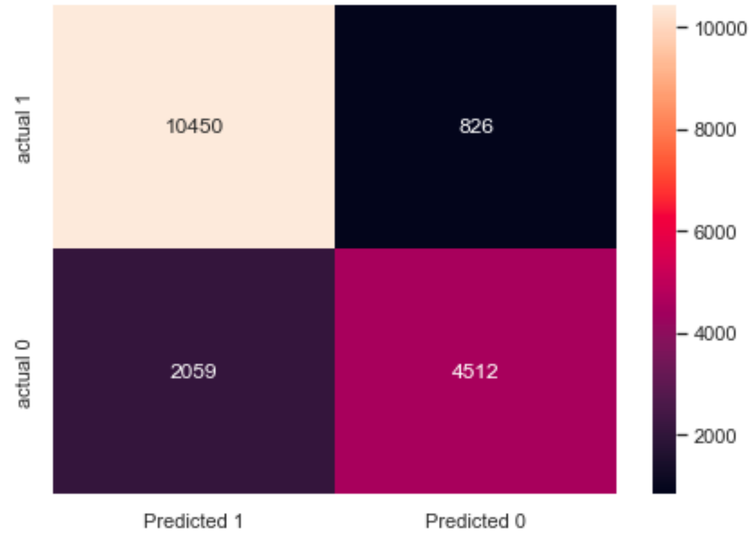
04. Model Building

02. Confusion Matrices



	precision	recall	f1-score	support
0	0.84	0.93	0.88	11276
1	0.85	0.69	0.76	6571
accuracy			0.84	17847
macro avg	0.84	0.81	0.82	17847
weighted avg	0.84	0.84	0.83	17847

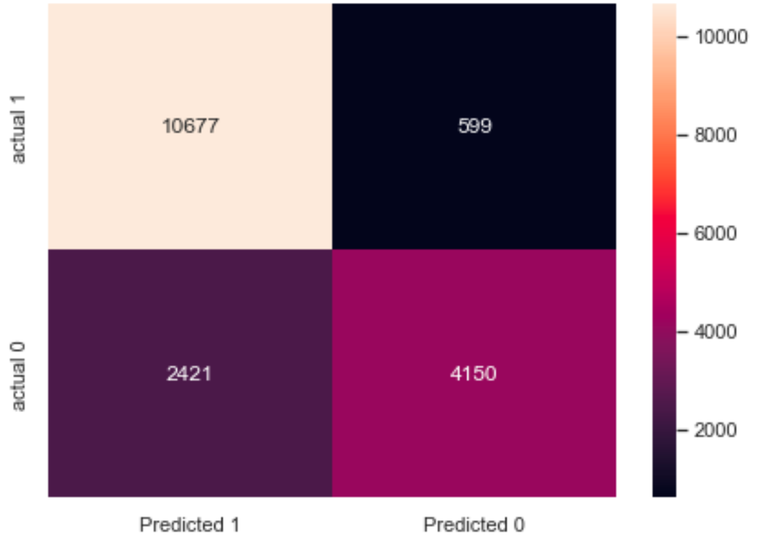
XGB Model



	precision	recall	f1-score	support
0	0.82	0.94	0.87	11276
1	0.85	0.64	0.73	6571
accuracy			0.83	17847
macro avg	0.84	0.79	0.80	17847
weighted avg	0.83	0.83	0.82	17847

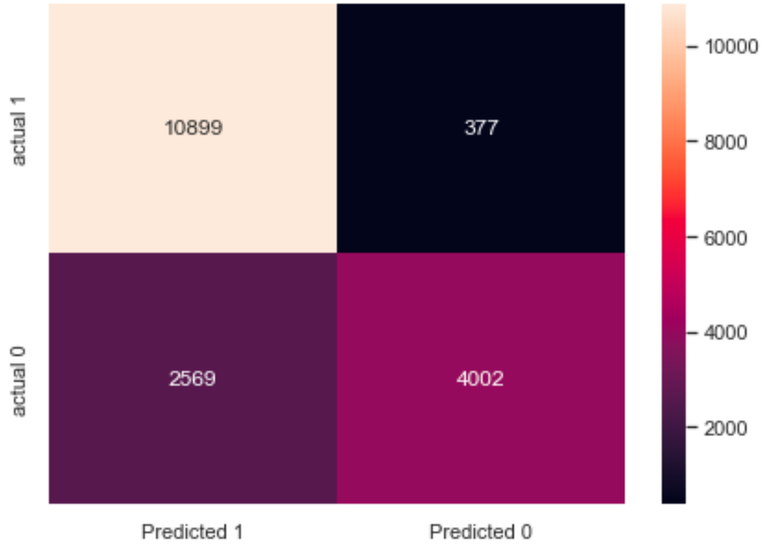
LGB Model

04. Model Building



	precision	recall	f1-score	support
0	0.82	0.95	0.88	11276
1	0.87	0.63	0.73	6571
accuracy			0.83	17847
macro avg	0.84	0.79	0.80	17847
weighted avg	0.84	0.83	0.82	17847

Random Forest Model



	precision	recall	f1-score	support
0	0.81	0.97	0.88	11276
1	0.91	0.61	0.73	6571
accuracy			0.83	17847
macro avg	0.86	0.79	0.81	17847
weighted avg	0.85	0.83	0.83	17847

Bagging Classifier Model

04. Model Building

03. Building Ensemble Model

Voting Classifier

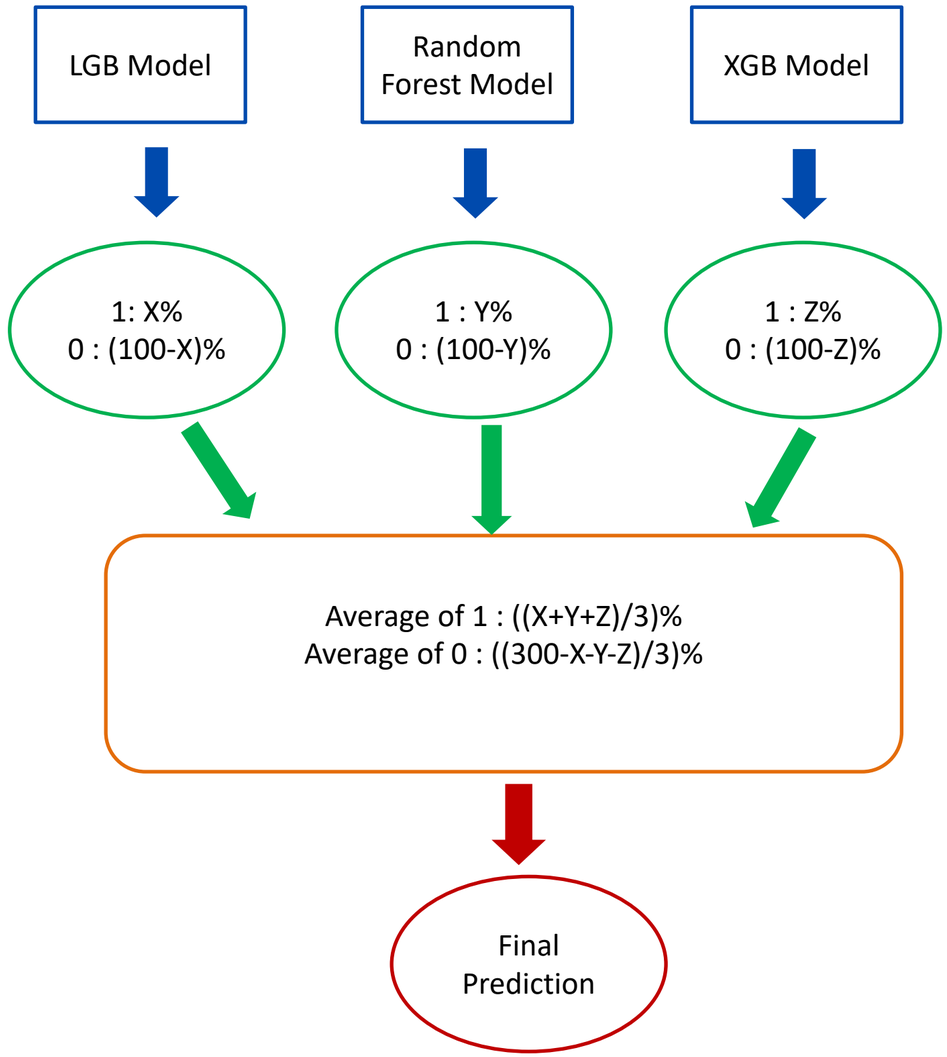
```
from sklearn.ensemble import VotingClassifier

rf = RandomForestClassifier(max_depth= None,n_estimators=300 )
lgbm = lgb.LGBMClassifier( max_depth= 7, n_estimators=100,num_leaves=32)
xgb= XGBClassifier( max_depth=9, n_estimators= 200)

voting_clf = VotingClassifier(estimators=[('lgbm', lgbm), ('rf', rf),('xgb',xgb)], voting='soft')
```

Cross Validation Accuracy Score : **85.30%**
Standard deviation of cross-validation scores : 0.0045

Cross Validation F1-Score : **78.34%**



04. Model Building

03. Building Ensemble Model

Stacking Classifier

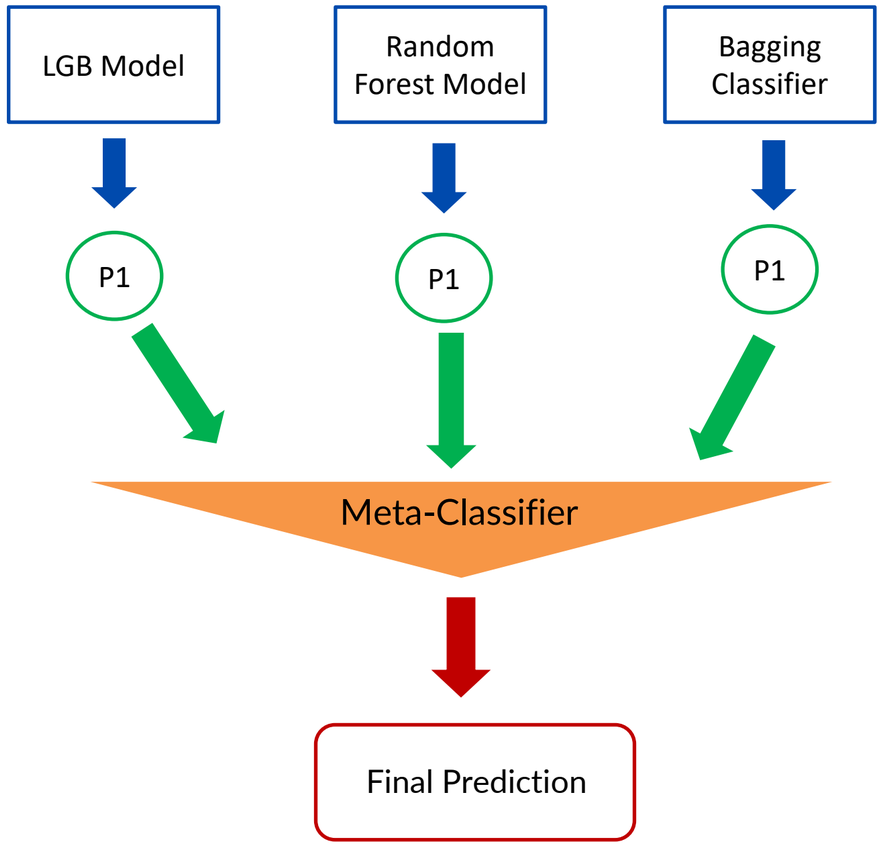
```
from sklearn.ensemble import StackingClassifier

rf_model = RandomForestClassifier(min_samples_split=6)
lgb_model=lgb.LGBMClassifier(learning_rate=0.01, max_depth=5, num_leaves=20,n_estimators=500)
bag_model = BaggingClassifier(base_estimator=DecisionTreeClassifier(max_depth=10, min_samples_split=5))


estimator_list = [
 ('rf_model' , rf_model),
 ('lgb_model' , lgb_model),
 ('bag',bag_model)

]

# Build stack model
stack_model = StackingClassifier(
    estimators=estimator_list, final_estimator=LogisticRegression(C=0.01,penalty="l2"), cv=5
)
```



Cross Validation Accuracy Score : **84.85%**
Standard deviation of cross-validation scores : 0.0049

Cross Validation F1-Score : **77.62%**

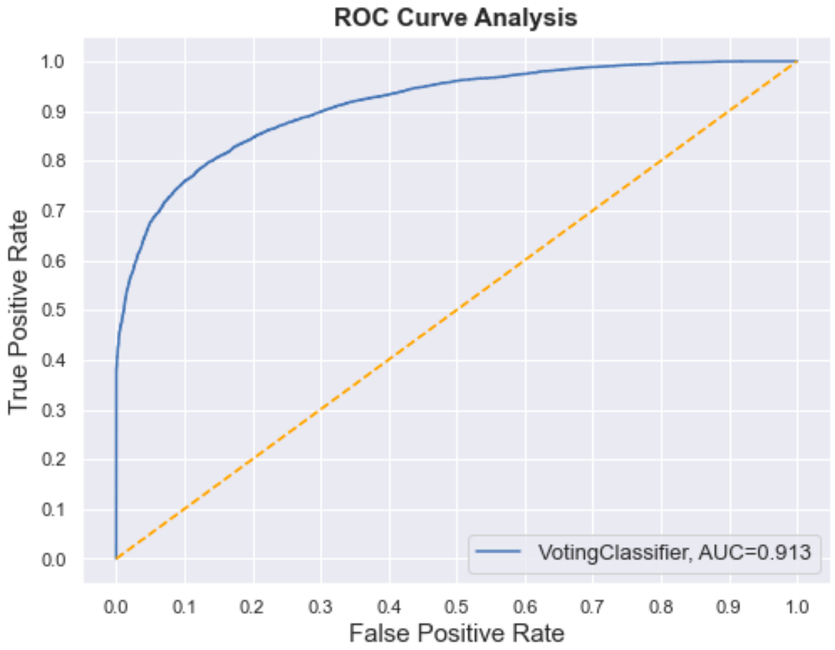
Stacking Classifier → **Higher Accuracy**
Higher F1–Score

04.Results

Stacking Classifier

Cross Validation Accuracy Score : **84.85%**
Standard deviation of cross-validation scores : 0.0049

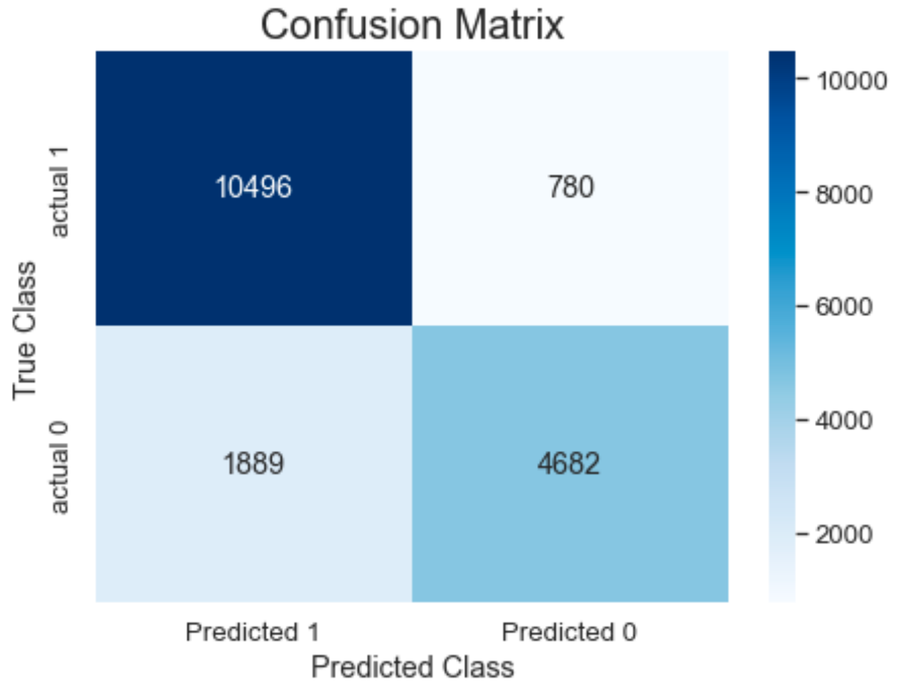
Cross Validation F1-Score : **77.62%**



ROC Curve of Stacking Classifier

Stacking Classifier Classification Report

	precision	recall	f1-score	support
0	0.85	0.93	0.89	11276
1	0.86	0.71	0.78	6571
accuracy			0.85	17847
macro avg	0.85	0.82	0.83	17847
weighted avg	0.85	0.85	0.85	17847



Stacking Classifier Confusion Matrix

Expose the Model as an API Endpoint

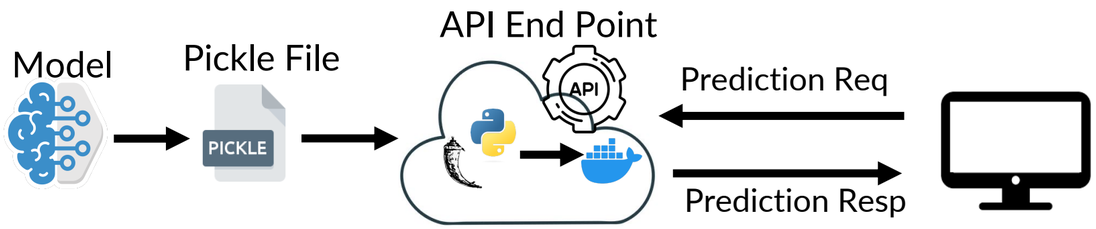
Exported the model into a pickle file

Created a Flask Application

Loaded the pickled model in Flask.

Developed an endpoint to receive requests.

Built the Docker image with Flask and the model



API Request :

GET http://127.0.0.1:5000/predict_cancellation Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{

"adr": 107,
"arrival_date_day_of_month": "2",
"arrival_date_month": "April",
"booking_changes": 0,
"customer_type": "Group",
"deposit_type": "No Deposit",
"lead_time": 203,
"market_segment": "Corporate",
"previous_cancellations": 1,
"required_car_parking_spaces": 1,
"total_of_special_requests": 1,
"total_stays": 2
}
```

Successful Response :

Body Cookies Headers (6) Test Results 200 OK 83 ms 287 B Save Response

Pretty Raw Preview Visualize JSON

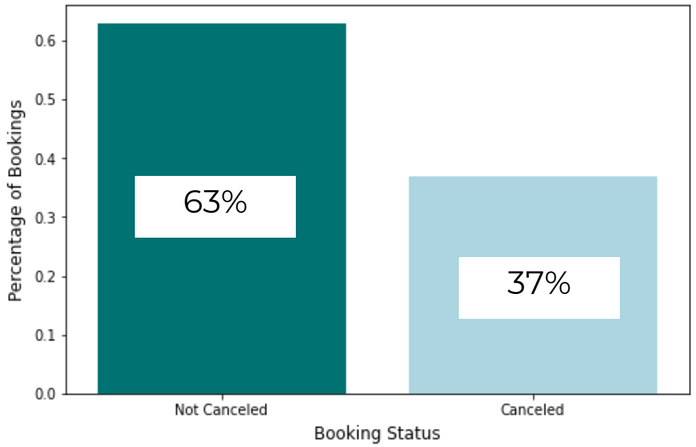
```
{
    "cancellation_probality": 8.97,
    "confirmation_probality": 91.03,
    "prediction": 0
}
```

Exploratory Data Analysis

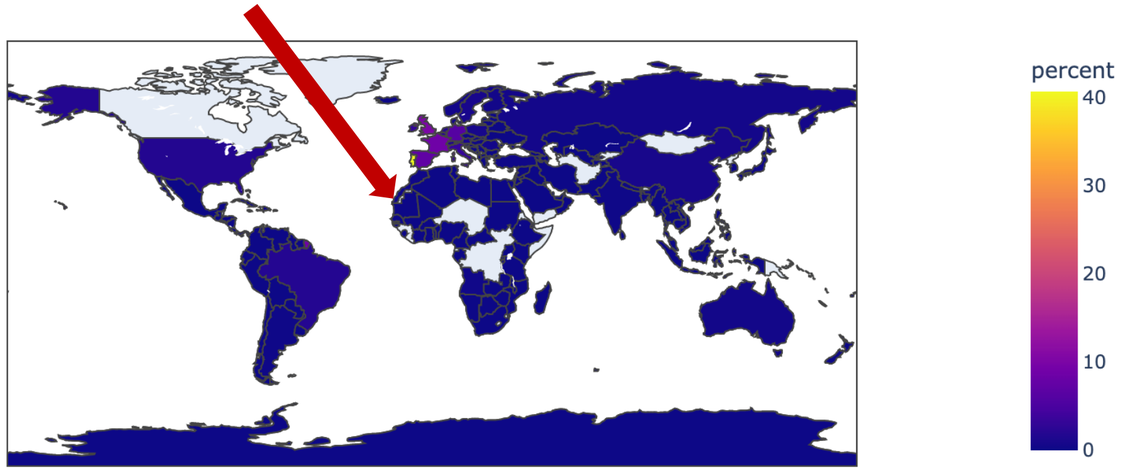
01. Univariate Analysis

```
df.describe()
✓ 0.1s
```

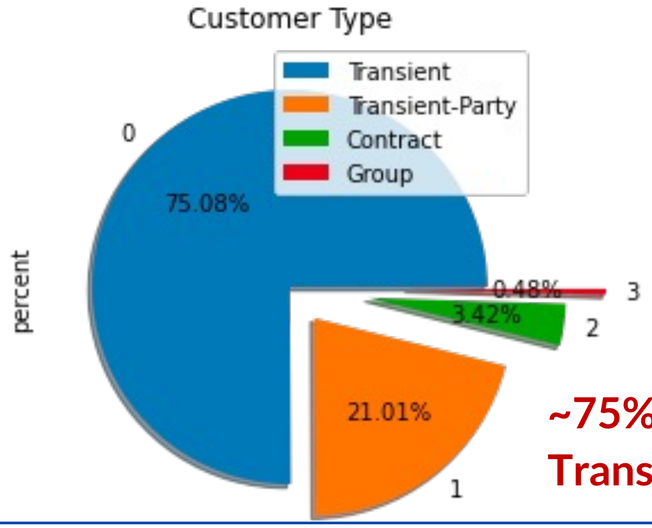
	is_canceled
count	118981.000000
mean	0.370731
std	0.483003
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000



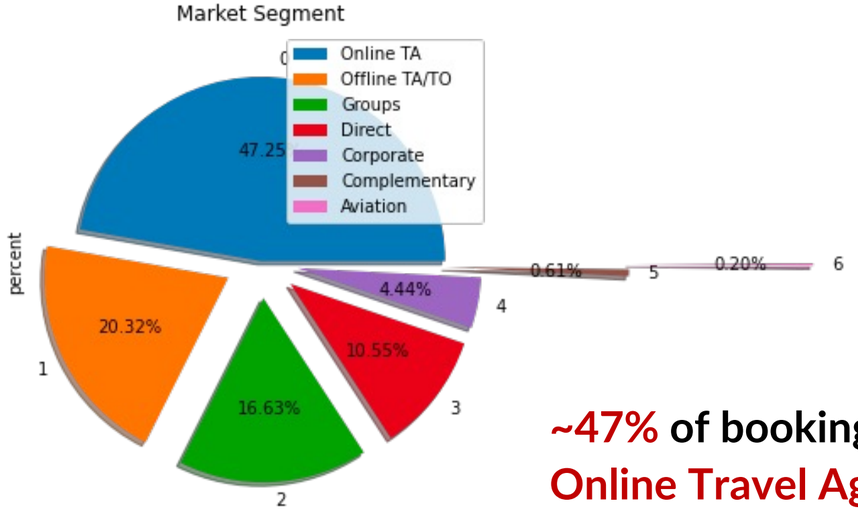
~40% customers home country was **Portugal**



Average cancellations are 37% and deviates by 48%. **Lots of variations !**



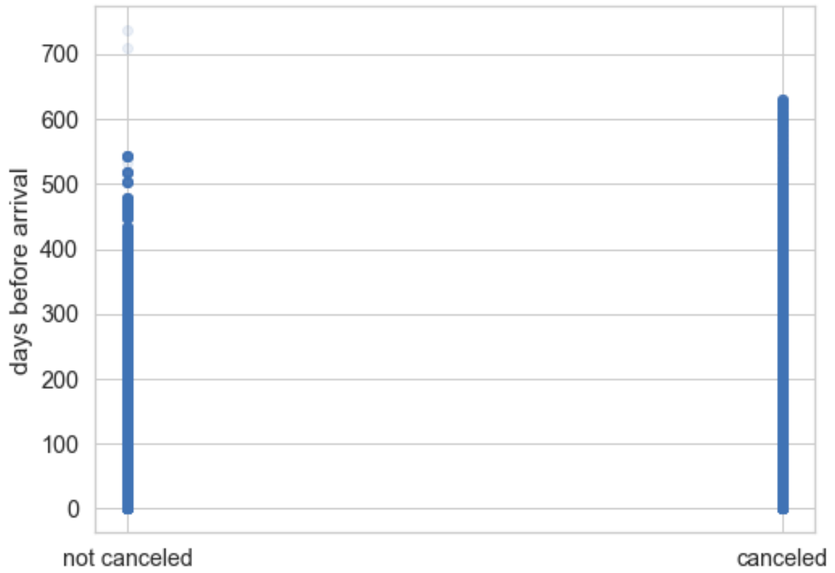
~75% Customers belongs to **Transient** type



~47% of booking were done from **Online Travel Agents**

Exploratory Data Analysis

02. Bi-Variate Analysis

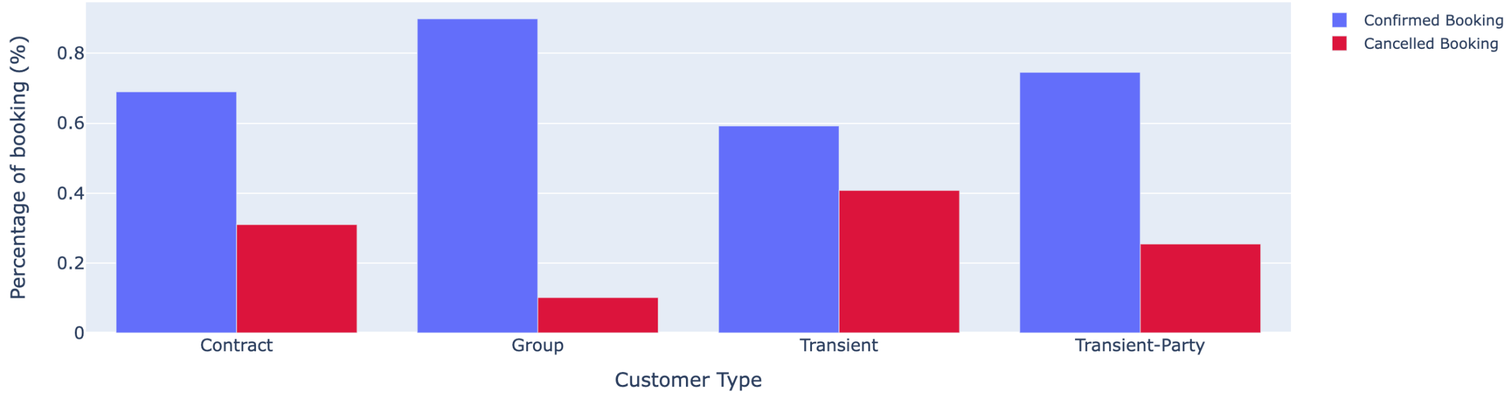


Longer lead time results in more **cancellations** !

Why does Non Refund Deposit Type Are More Likely to Cancelled ?

Longer lead time

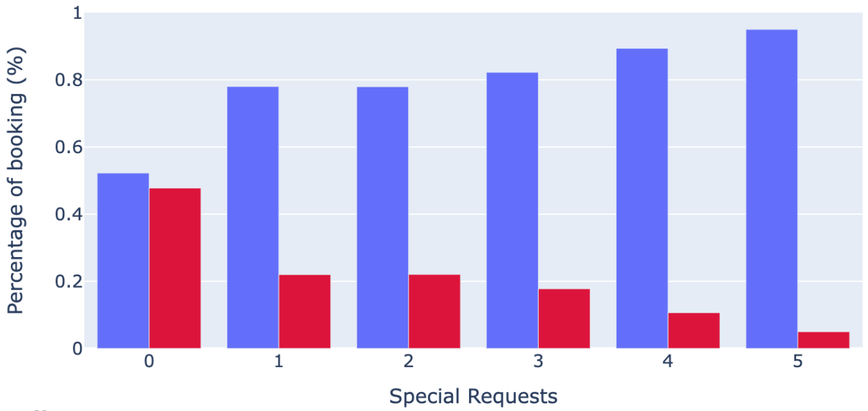
col_0	Median Lead Time
deposit_type	
No Deposit	56
Non Refund	183
Refundable	169



Customers who comes in **Groups** are **less likely to Cancel**

Exploratory Data Analysis

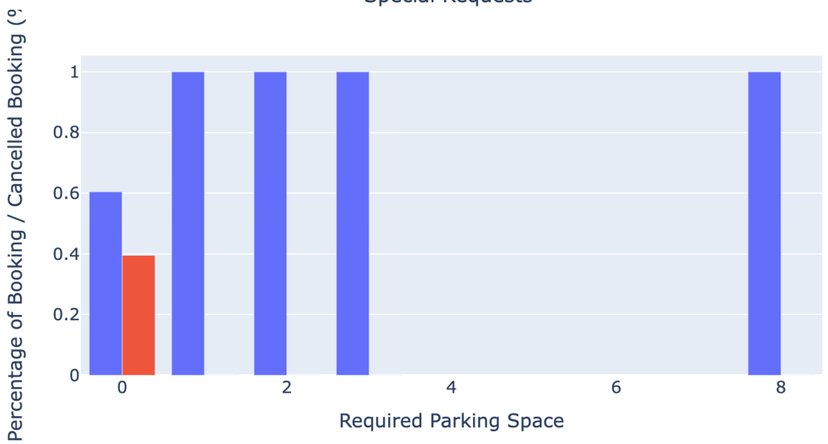
02. Bi-Variate Analysis



Booking that has no special request are more likely to cancelled

is_canceled	0	1
is_previously_cancelled		
0	0.660705	0.339295
1	0.083218	0.916782

Booking that's been previously cancelled are more likely to be cancelled again !



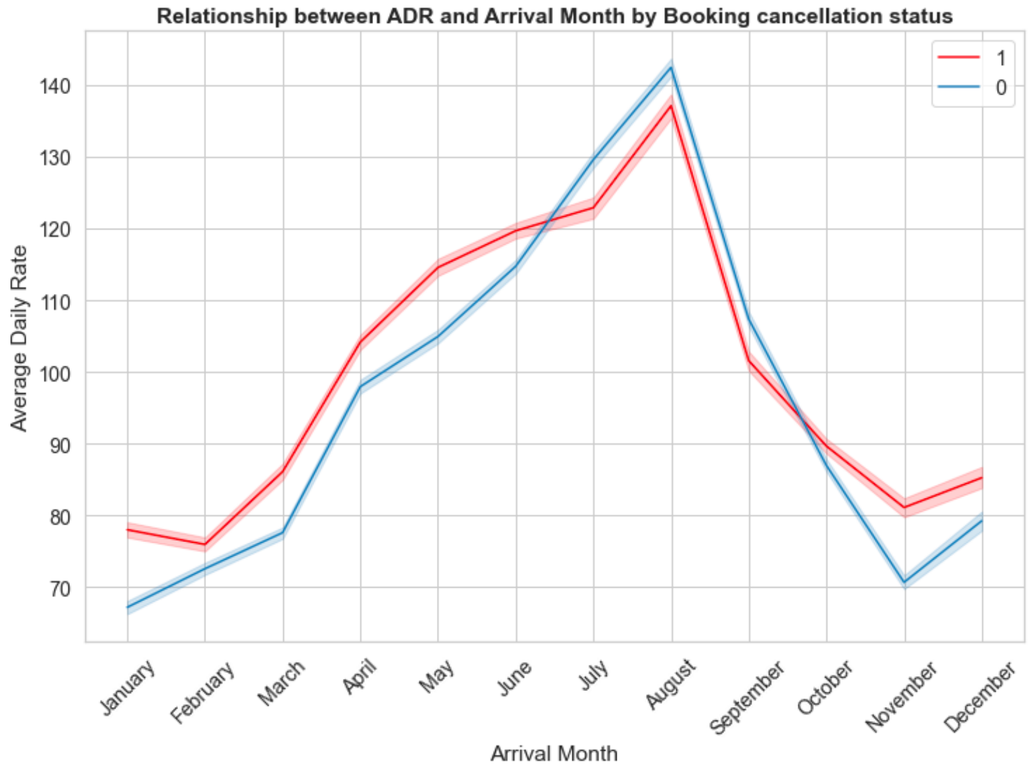
Bookings required a parking space will high likely to be confirmed

is_canceled	0	1
is_booking_changes		
0	0.591399	0.408601
1	0.842848	0.157152

Customer Who made booking Changes to their booking have a lower cancellation rate

Exploratory Data Analysis

03. Multi-Variate Analysis



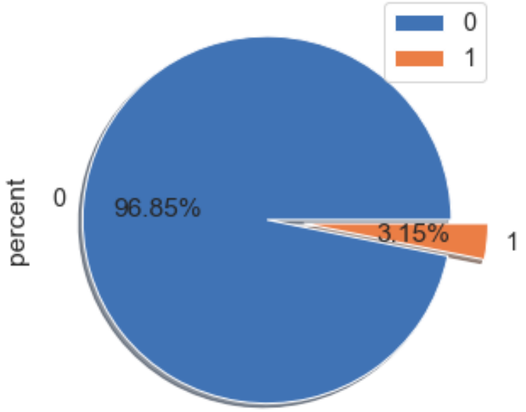
Seasonality Trends

- A greater number of bookings were made in moth of august
- Number of cancellation generally goes up as the number of booking goes up

Exploratory Data Analysis

04. Customer Loyalty

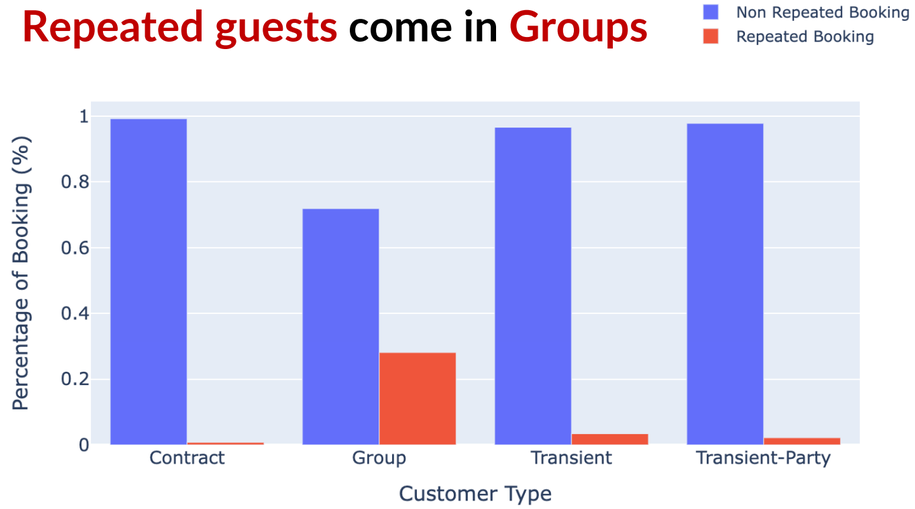
~3% of Repeated Guests



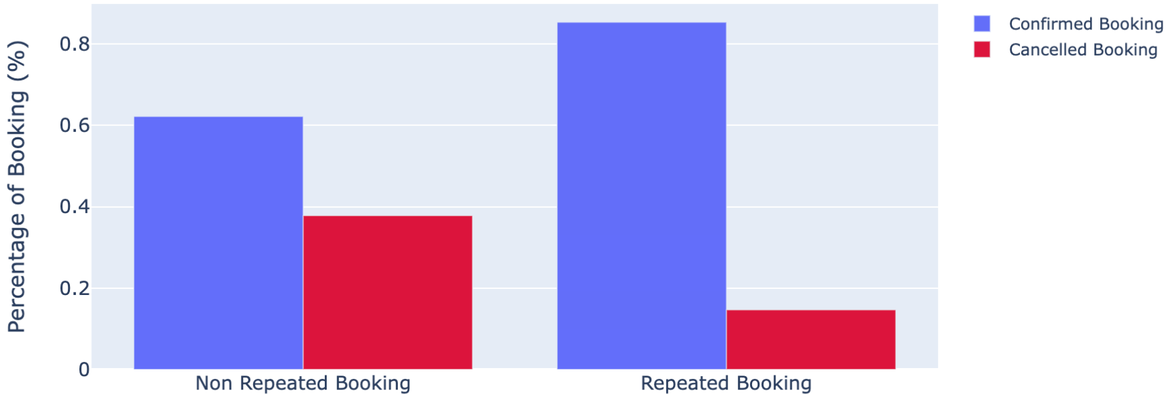
Repeated guests dissatisfied

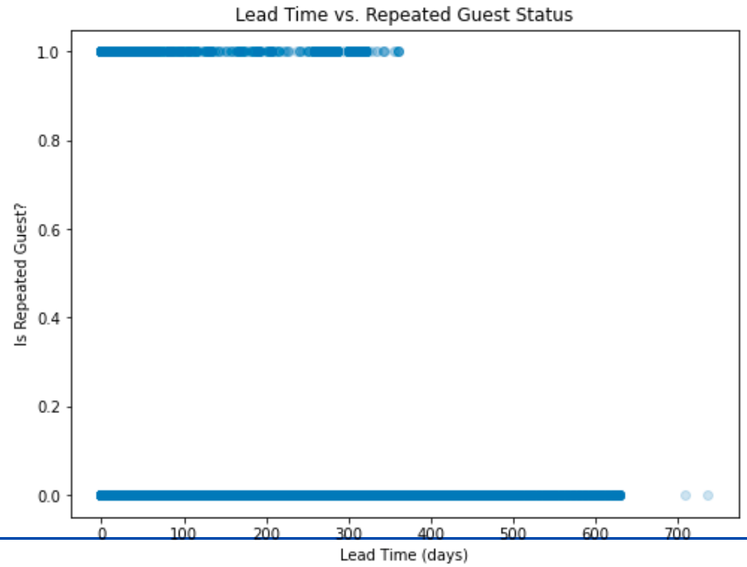
stays_group	is_repeated_guest	total_stays
1–3	No	0.373529
	Yes	0.012109
4–7	No	0.466494
	Yes	0.003246
8+	No	0.142049
	Yes	0.002573

Repeated guests come in Groups



Non Repeated Guest are more than 2X more likely to cancel





Repeated guests have less Lead Time

06. References

[1]Condor Ferries, “65+ Hotel Industry Statistics & Trends (2020),” *Condor Ferries*, 2020. https://www.condorferries.co.uk/hotel-industry-statistics (accessed Mar. 11, 2023).

[2]“Hotel cancelation rate at 40% as online travel agencies push free change policy,” *www.phocuswire.com*. https://www.phocuswire.com/Hotel-distribution-market-share-distribution-analysis#:~:text=The%20average%20cancelation%20rate%20in (accessed Mar. 10, 2023).

[3]T. J. McCue, “Stop Losing Money When You Have To Cancel Hotel Reservations,” *Forbes*. https://www.forbes.com/sites/tjmccue/2014/03/31/stop-losing-money-when-you-have-to-cancel-hotel-reservations/?sh=78ee51037ce3 (accessed Mar. 10, 2023).

[4]N. Antonio, A. de Almeida, and L. Nunes, “Predicting hotel booking cancellations to decrease uncertainty and increase revenue,” *Tourism & Management Studies*, vol. 13, no. 2, pp. 25–39, Apr. 2017, doi: https://doi.org/10.18089/tms.2017.13203.

[5]N. Antonio, A. de Almeida, and L. Nunes, “An Automated Machine Learning Based Decision Support System to Predict Hotel Booking Cancellations,” *Data Science Journal*, vol. 18, 2019, doi: https://doi.org/10.5334/dsj-2019-032.

[6]Md. S. Satu, K. Ahammed, and M. Z. Abedin, “Performance Analysis of Machine Learning Techniques to Predict Hotel booking Cancellations in Hospitality Industry,” *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, Dec. 2020, doi: https://doi.org/10.1109/iccit51783.2020.9392648.

[7]M. Adil, M. F. Ansari, A. Alahmadi, J.-Z. Wu, and R. K. Chakrabortty, “Solving the Problem of Class Imbalance in the Prediction of Hotel Cancelations: A Hybridized Machine Learning Approach,” *Processes*, vol. 9, no. 10, p. 1713, Sep. 2021, doi: https://doi.org/10.3390/pr9101713.

06. Conclusion

1. Hotel cancelation models = less risk + more sales.
2. Models help managers prepare for cancellations = better decisions.
3. Machine learning improves accuracy = benefits for hotels and customers.

THANK YOU !

Do You Have Any Questions?