KYP X TFT (DATA SCIENCE ANALYTICS)

PREDICTION OF HOTEL BOOKING CANCELLATION

Muhammad Farhan Hafizi bin Abdul Rodzi
Nurul Abidah binti Shukor
Puteri Raifeeza binti Ahmad Zai

8th August 2024

ABOUT US

Farhan Hafizi
Future Data Analyst

- Former Quran teacher and crew restaurant
- Bachelor of Islamic Studies (Syariah) with Honours

Abidah Shukor
Future Data Analyst

- Former Marketing Officer @Farmasi Muslim Pekan
- Bachelor of Quranic and Sunnah with Honours

Puteri Raifeeza
Future Data Analyst

- A home-based businesswoman @ florist
- Bachelor of Chemical Engineering @ UM

PEOPLElogy
DISCOVERY • DEVELOPMENT • DIGITALISATION

LIST OF CONTENTS

COURSE JOURNEY

DATA SCIENCE ANALYTICS BOOTCAMP

WEEK 1

- Self Resilience
- Communication
- Career Launchpad Mastery

WEEK 2

- Big Data and Data Science
- Data Science Methodology
- Python Programming

WEEK 3

- Data Wrangling
- Data Visualisation
- Exploratory Data Analysis

WEEK 4

- Machine Learning
- Deep Learning
- MySQL

BUSINESS UNDERSTANDING

This dataset contains booking informations for City Hotel and Resort Hotel.

It includes informations such as when the booking was made, length of stays, the number of adults, children, and/or babies, platform of hotel booking and among other things.

BUSINESS UNDERSTANDING

Problems

Guest cancellation behavior
(last minute, personal circumstances)

Seasonal Factors Affecting Cancellations

Objective

To predict cancellation of hotel bookings using machine learning algorithms.

Main Goal

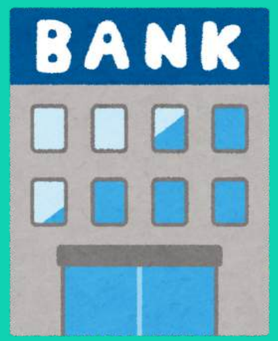
Choose the best model in predicting cancellation of hotel bookings.

ANALYTIC FRAMEWORK

Data collection &
Data wrangling

Model selection &
Development

Model
Selection &
Deployment



Exploratory Data
Analysis (EDA)

Model
Evaluation

ANALYTIC APPROACH

Type of Machine Learning :
Supervised Learning

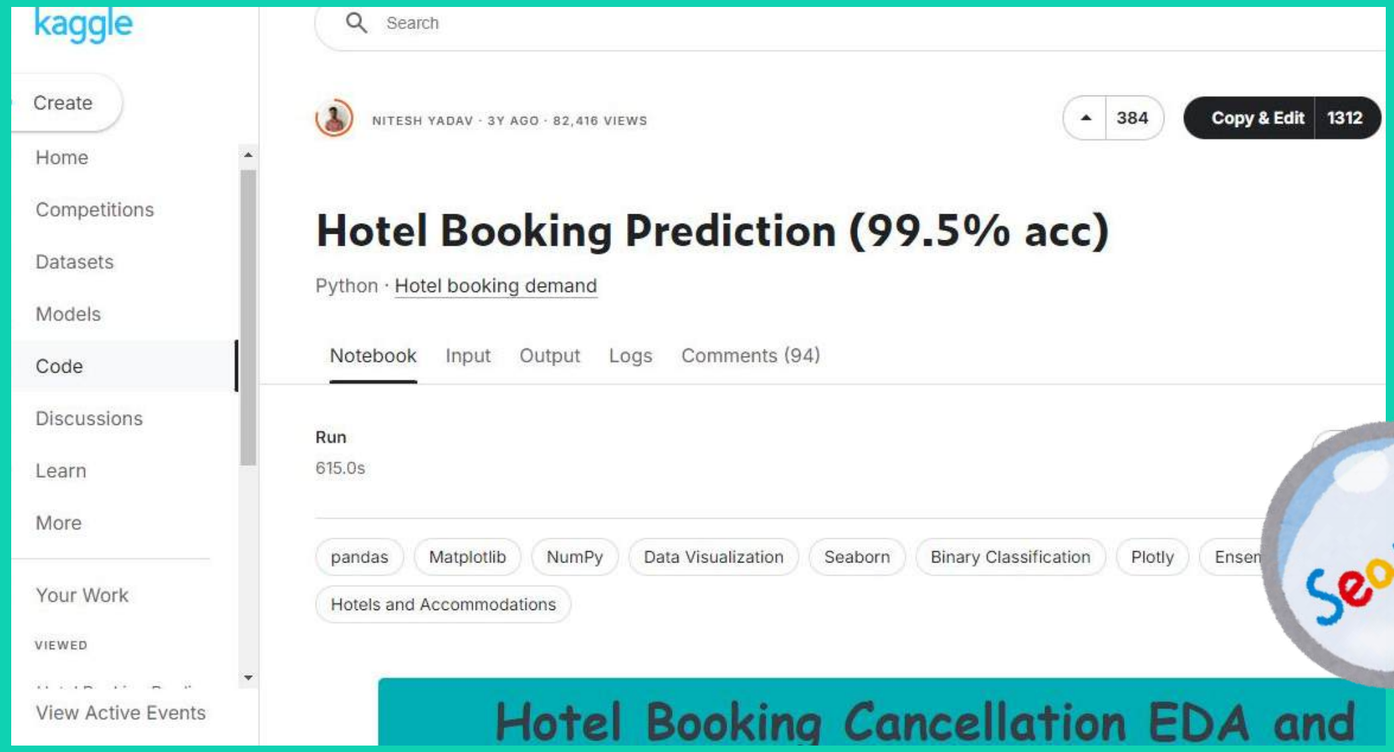
Classification

- K-Nearest Neighbours
- Decision Tree
- Random Forest
- XG Boost
- Logistic Regression

DATA REQUIREMENTS

Link:

https://www.kaggle.com/code/niteshyadav3103/hotel-booking-prediction-99-5-acc



Data is from **Kaggle**.

DATA COLLECTION

Data sources

The dataset consists of 119,390 records with the following attributes:

- hotel (type of 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_guest**
- previous_cancellations
- previous_bookings_not_canceled
- reserved_room_type
- assigned_room_type
- booking_changes
- **deposit_type**
- agent
- company
- days_in_waiting_list
- **customer_type**
- adr (average daily rate)
- **required_car_parking_spaces**
- **total_of_special_requests**
- reservation_status
- reservation_status_date

DATA UNDERSTANDING

Load '/content/hotel_bookings.csv' into DataFrame

```
# reading data
df = pd.read_csv('/content/hotel_bookings.csv')
df.head()
```

	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_we
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
2	Resort Hotel	0	7	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

Descriptive statistics for a DataFrame

```
[ ] df.describe()
```

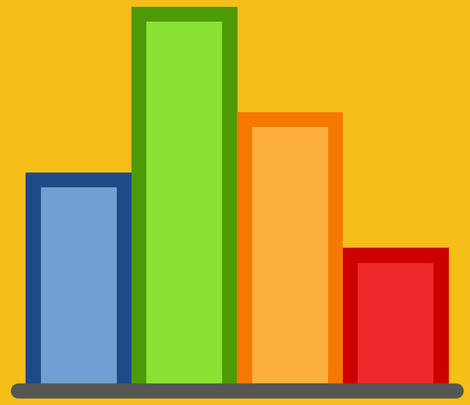
	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000

DATA UNDERSTANDING

Identify missing values.

```
# Filter features with missing values
missing_features = missing_values[missing_values > 0]
print(missing_features)
```

```
children         4
country        488
agent        16340
company     112593
dtype: int64
```



Finding correlations between features

```
[15] #select only numerical columns
df_numeric = df.select_dtypes(include=['number'])

#generate correlation
df_numeric.corr()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	a
is_canceled	1.000000	0.293123	0.016660	0.008148	
lead_time	0.293123	1.000000	0.040142	0.126871	
arrival_date_year	0.016660	0.040142	1.000000	-0.540561	
arrival_date_week_number	0.008148	0.126871	-0.540561	1.000000	
arrival_date_day_of_month	-0.006130	0.002268	-0.000221	0.066809	
stays_in_weekend_nights	-0.001791	0.085671	0.021497	0.018208	
stays_in_week_nights	0.024765	0.165799	0.030883	0.015558	
adults	0.060017	0.119519	0.029635	0.025909	
children	0.005048	-0.037622	0.054624	0.005518	
babies	-0.032491	-0.020915	-0.013192	0.010395	
is_repeated_guest	-0.084793	-0.124410	0.010341	-0.030131	
previous_cancellations	0.110133	0.086042	-0.119822	0.035501	

DATA UNDERSTANDING

Identify data type of each columns :
object / integer / float

```
[ ]  df.info()

<class 'pandas.core.frame.DataFrame'>
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
```

DATA UNDERSTANDING

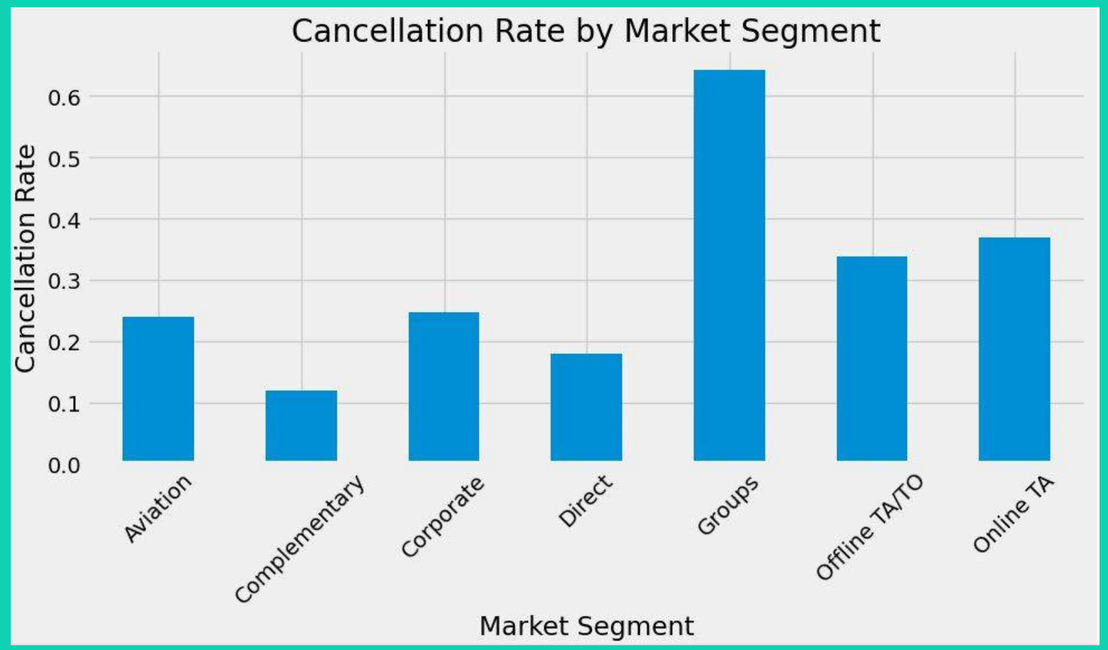
Identify categorical and numerical features

```
[ ]  # Identify categorical and numerical features
     categorical_features = df.select_dtypes(include=['object']).columns
     numerical_features = df.select_dtypes(include=['number']).columns

     print("Categorical Features:", categorical_features)
     print("Numerical Features:", numerical_features)
```

```
Categorical Features: Index(['hotel', 'arrival_date_month', 'meal', 'country', 'market_segment',
       'distribution_channel', 'reserved_room_type', 'assigned_room_type',
       'deposit_type', 'customer_type', 'reservation_status',
       'reservation_status_date'],
      dtype='object')
Numerical Features: Index(['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_bookings_not_canceled', 'booking_changes', 'agent',
       'days_in_waiting_list', 'adr', 'required_car_parking_spaces',
       'total_of_special_requests'],
      dtype='object')
```

DATA PREPARATION

Bar Plot

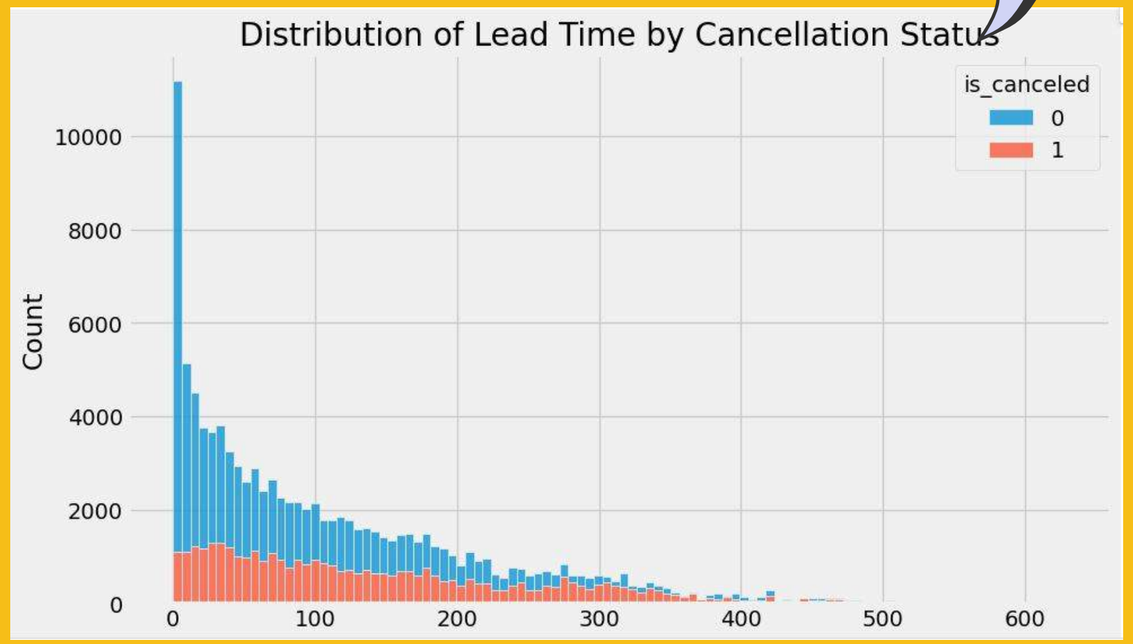


Market Segment VS Cancellation Status

Groups in Market Segment has the highest cancellation rate.

Histplot

Lead time:
Number of days between booking date to arival date

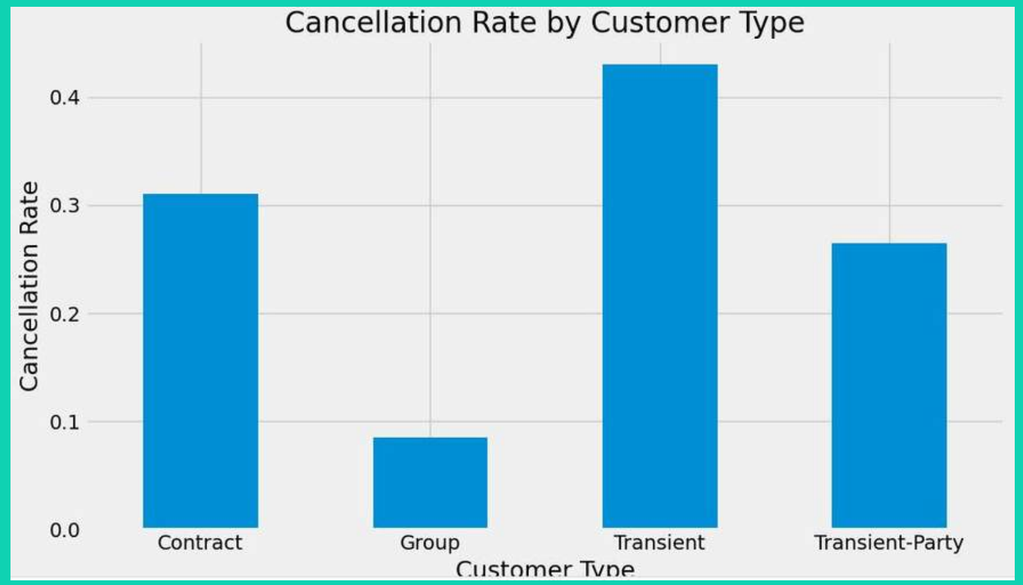


Lead Time VS Cancellation Status

As the lead time is increasing, the cancellation status is decreasing.

DATA PREPARATION

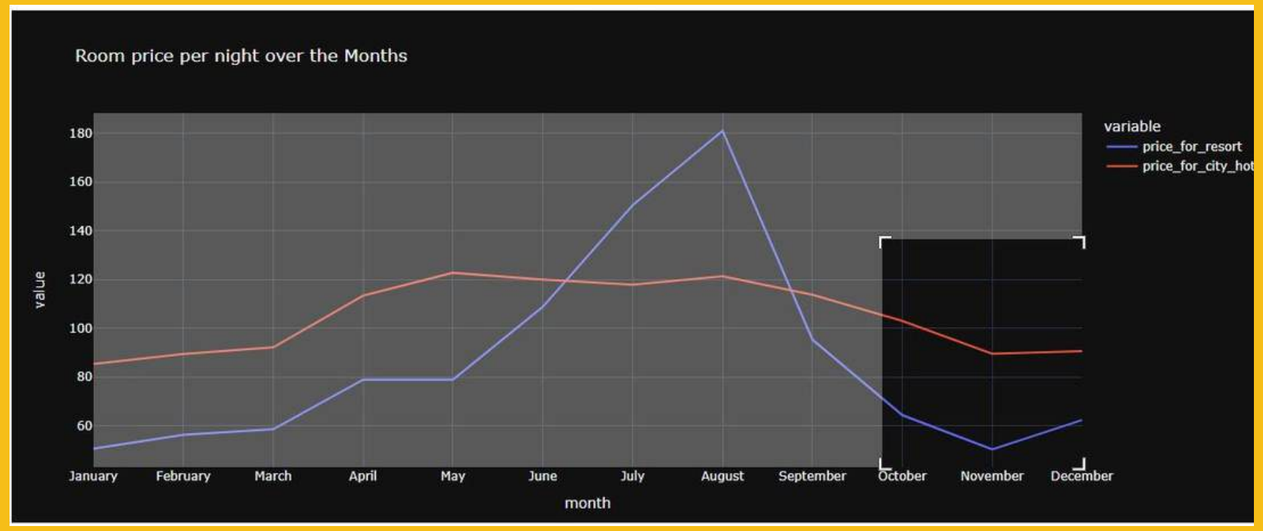
Bar Plot



Customer Type vs Cancellation Rate

Transient in Customer Type has the highest cancellation rate.

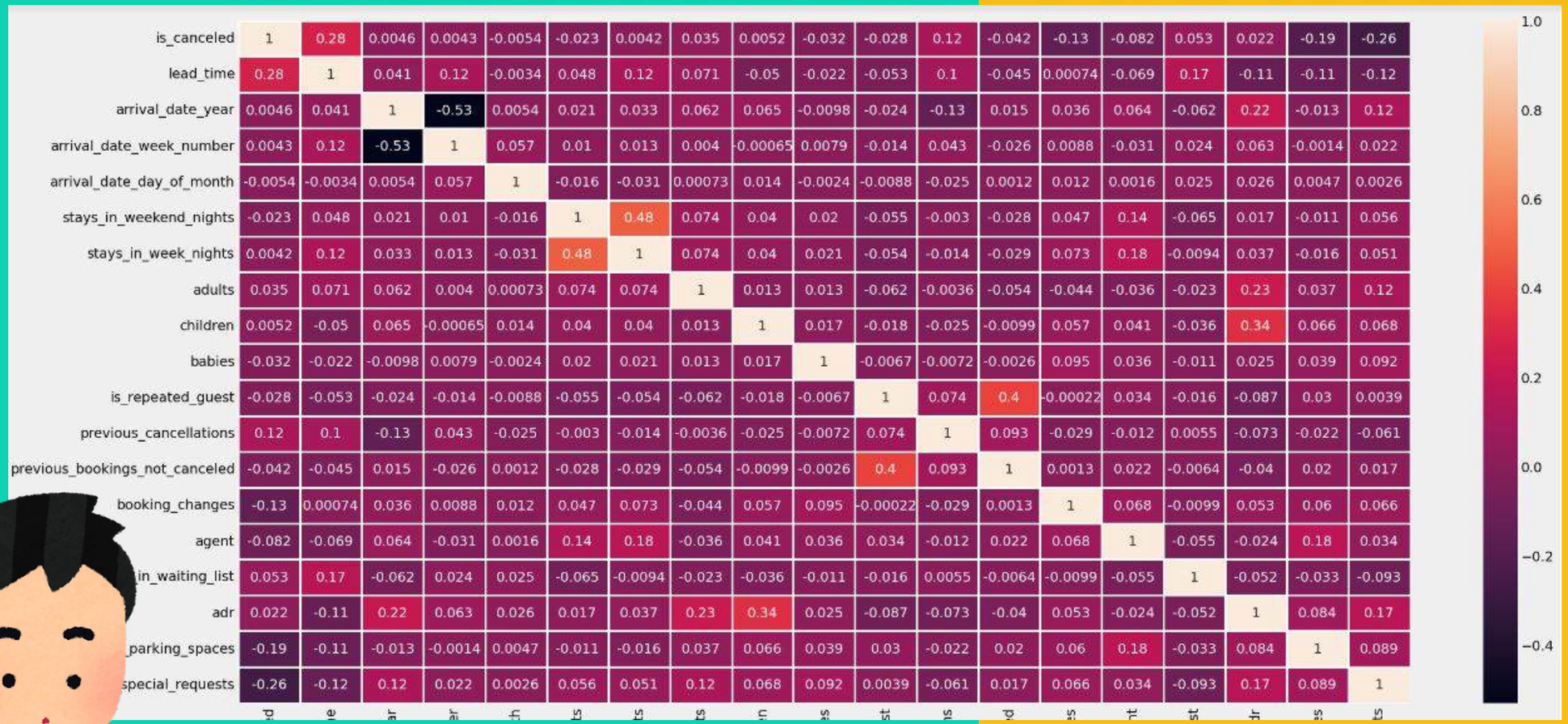
Line Plot



Month vs Value

- Prices in the Resort Hotel are much higher during the summer.
- Prices of City hotel varies less and is most expensive during Spring and Autumn

DATA PREPARATION



Correlation table

is_canceled	1
lead_time	0.28
arrival_date_year	0.0046
arrival_date_week_number	0.0043
arrival_date_day_of_month	-0.0054
stays_in_weekend_nights	-0.023
stays_in_week_nights	0.0042
adults	0.035
children	0.0052
babies	-0.032
is_repeated_guest	-0.028
previous_cancellations	0.12
previous_bookings_not_canceled	-0.042
booking_changes	-0.13
agent	-0.082
days_in_waiting_list	0.053
adr	0.022
required_car_parking_spaces	-0.19
total_of_special_requests	-0.26

x-axis features
hotel
market_segment
distribution_channel
deposit_type
customer_type
lead_time
is_repeated_guest
required_car_parking_spaces
total_of_special_requests

FEATURES SELECTION

1. **ANOVA (F-Score & P-score)**
2. **Correlation Matrix**

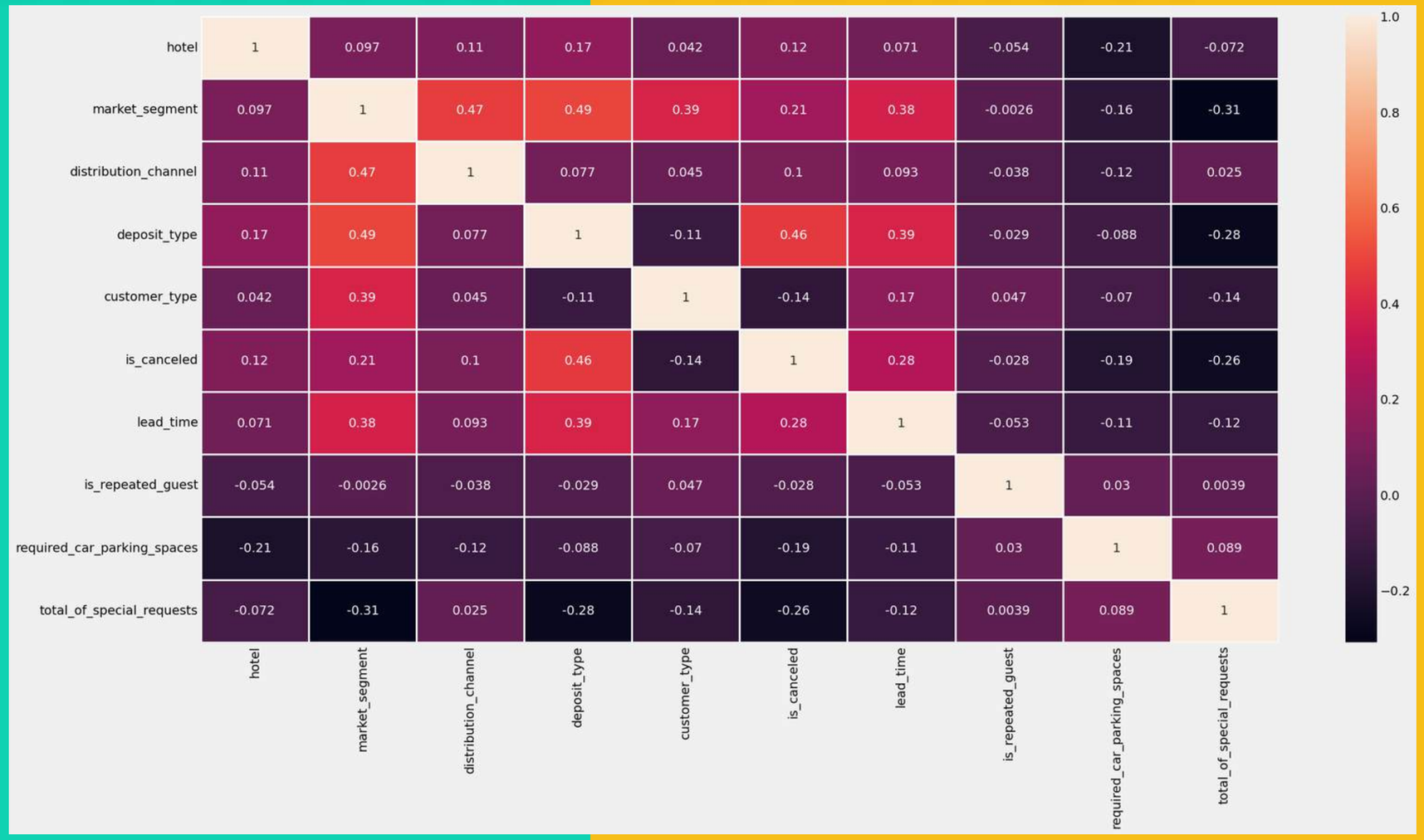
y-axis feature
is_canceled

DATA PREPARATION

Correlation table

with

selected features



MODELLING

a) Creating Train and Test Datasets

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

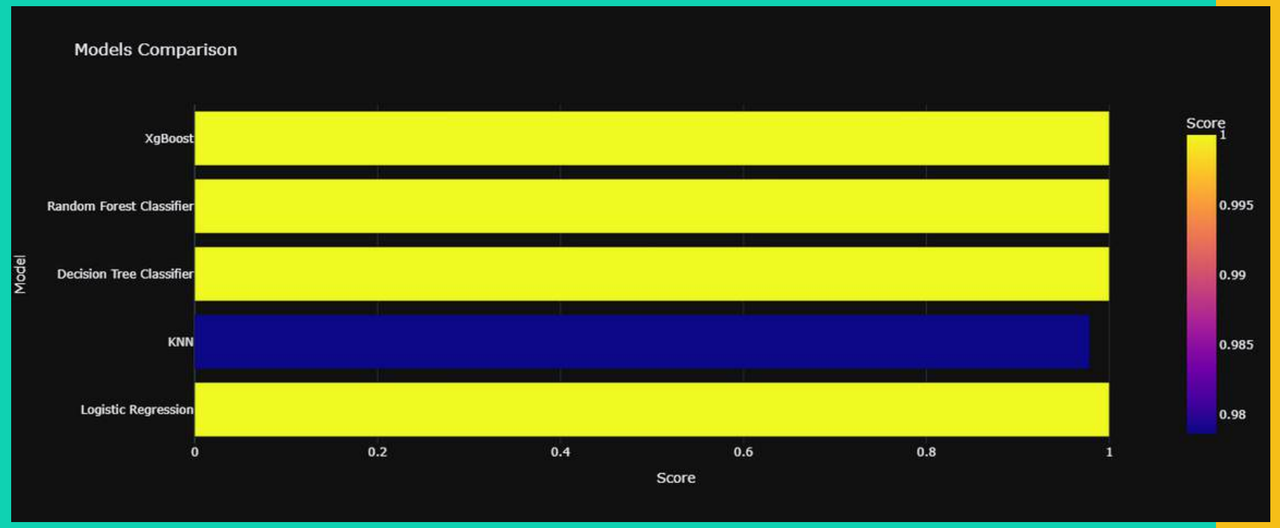
```
# splitting data into training set and test set

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

Training 70% Testing 30%

COMPARISON WITHIN MODELS

Model	Training Accuracy	Testing Accuracy
K- Nearest Neighbor	98.88%	98.00%
Decision Tree	100.00%	100.00%
Random Forest	100.00%	100.00%
XG Boost	100.00%	100.00%
Logistic Regression	100.00%	100.00%

EVALUATION



CROSS VALIDATION

To ensure that a model performance well on unseen data and is not overly fitted to the training data.

- **Estimate Model Performance**
- **Optimize Hyperparameters**
- **Reduce Overfitting**

Cross Validation for XGBoost

```
[ ]  #cross validation xgboost
     from sklearn.model_selection import cross_val_score

     scores = cross_val_score(xgb_clf, X, y, cv=5)  # 5-fold cross-validation

     print("Cross-validation scores:", scores)
     print("Average cross-validation score:", scores.mean())
```

```
Cross-validation scores: [1. 1. 1. 1. 1.]
Average cross-validation score: 1.0
```

EVALUATION

Cross Validation	**Accuracy %**
XGboost	100
Logistic Regression	99
Decision Tree	100
Random Forest	100
KNN	99

b) Hyperparameter Tuning

max_depth = 7
learning_rate = 0.2296
n_estimators = 120

Training Accuracy = 100%
Testing Accuracy = 100%

XGBOOST!!

CONCLUSION

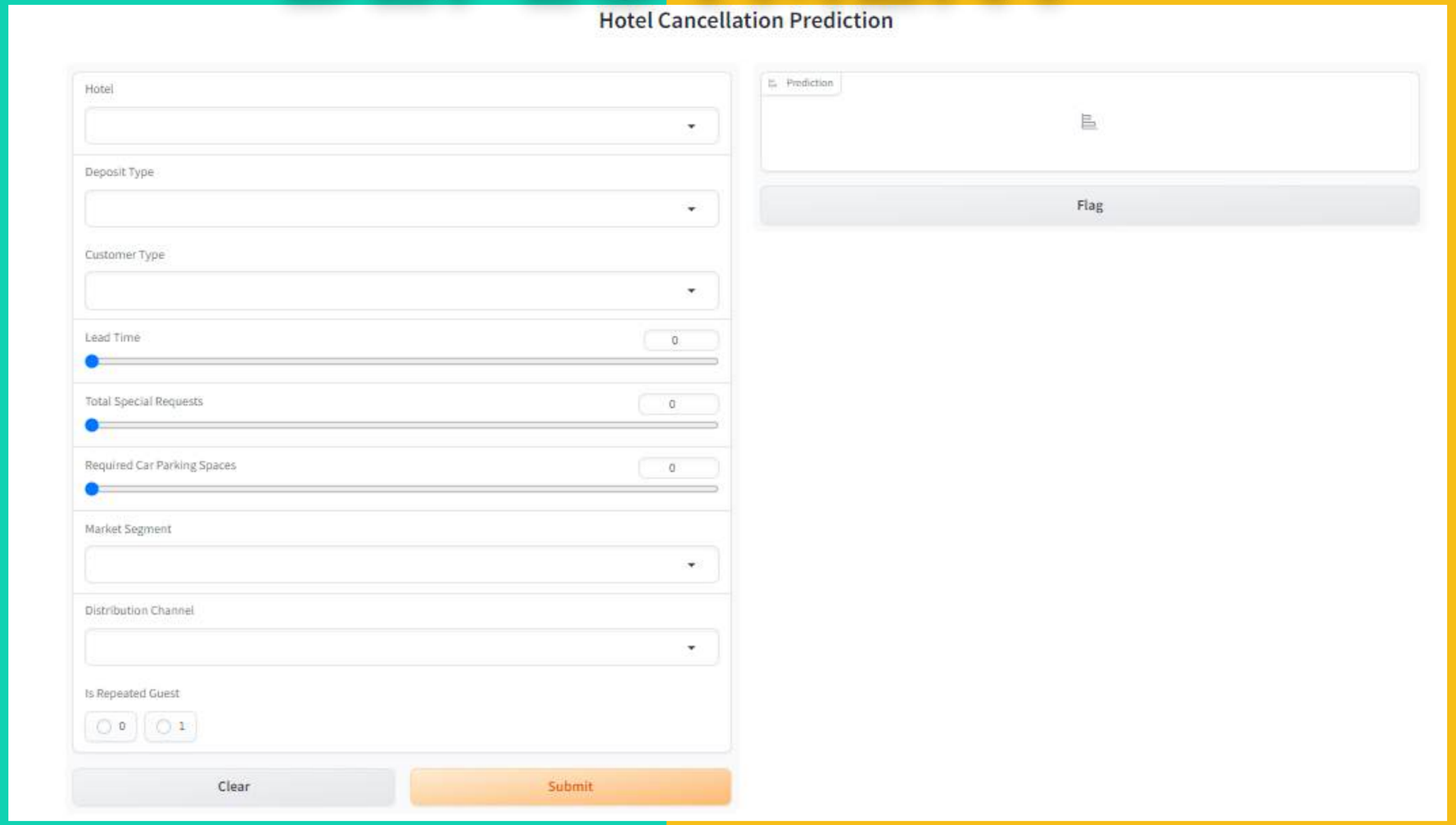
Objectives Achieved

- Cleaned and visualized data
- Identified the suitable features that significant to prediction of hotel booking cancellations
- Managed to choose the best model with highest accuracy and performance

Future Steps

- Confirm Accuracy
- Test Set Evaluation

DEPLOYMENT



https://b628ebf6529e3164aa.gradio.live/

PROJECT BUDGET

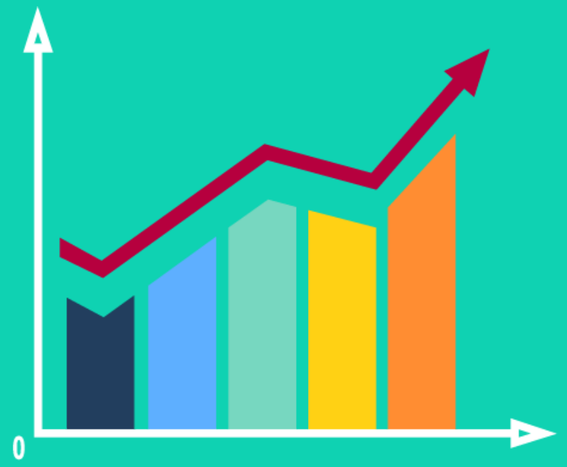
Initial cost

Labour	Monthly Time Spent	Number of Month	Head Count	Subtotal
Data Analyst	**40%**	**1**	2	**RM 4,800**
Data Scientist	**40%**	**1**	**1**	**RM 2000**
Total				**RM 6800**

THANK YOU

Q&A

THANKS FOR YOUR ATTENTION



FEEDBACK

CTA (CALL TO ACTION)

Linkedin : www.linkedin.com/in/farhan-hafizi-1363872ab
Github : https://github.com/Farhan-Hafizi
Email : farhanhafizi7012@gmail.com

Linkedin : www.linkedin.com/in/nurul-abidah-shukor-b105a0178
Github : https://github.com/Abidah15
Email : abidahshukor98@gmail.com

Linkedin : https://www.linkedin.com/in/puteri-raifeeza-17a032184/
Github : https://github.com/PuteriRaifeeza
Email : puteriraifeeza99@gmail.com