Hotel Cancellation Prediction

<u>G1- Group 1</u>

Anna Choo Xin Yi

Wesley Djingga

Xie Jianlong

Filbert

Zhang Jieyuan

Agenda

Item	Presenter
Business Problem	Anna
Dataset & Preparation	Anna
EDA	Anna
Featured Engineering	Wesley
Model and its hyper parameter	Wesley & Jieyuan
Model comparison and evaluation	Jianlong
Ensemble	Filbert
Conclusion	Filbert

Business Problem

Hotel Goal:

Overbooking as revenue management practice to minimize losses from late cancellations and no-shows (Kimes and Chase, 1998)

Problem:

Repercussions of this revenue management strategy

Repercussions

1. Compensation incurred

today), we found that approximately 30 percent of participants also expected a free night or discounted stay at the original hotel at a later time in order to ensure their ongoing patronage (see Table VI). In addition, a complementary meal at the hotel was mentioned by 12 percent of participants. Furthermore, 14 percent of participants pointed to the quality of the hotel they were walked to as important. They expected the hotel to be the same as or nicer than the

2. Bad customer experience

As previously mentioned, it is widely believed that the use of revenue management practices may alienate customers owing to perceived unfairness, thus leading to decreased customer satisfaction and goodwill and, ultimately, to a loss in customer loyalty (Kahneman *et al.*, 1986a; Kimes, 1994; Wirtz *et al.*, 2002) and long-run profits (Kimes, 2002). Yet, the behavioral

3. Bad reputation

consequences, Blodgett *et al.* (1997) found that people who perceived injustice were more likely to exhibit anger toward, engage in negative word-of-mouth publicity about, and detach themselves from the service provider perceived as unjust. In their model of the determinants

Literature review: The effect of perceived fairness toward hotel overbooking and compensation practices on customer loyalty (Hwang, J. and Wen, L., 2009)

Business Proposal

Aim:

Predict hotel booking cancellation using ML with customers' booking information.

Pros:

Analyze accurately in advance how many rooms can be overbooked

1) Resolve repercussions
2) Maximize revenue via improve occupancy
3) Better customer experience and reputations

Data Preparation

Dataset:

Hotel Booking Demand Datasets

~ written by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019

Consist:

2 separate files (119,390 rows and 31 columns)

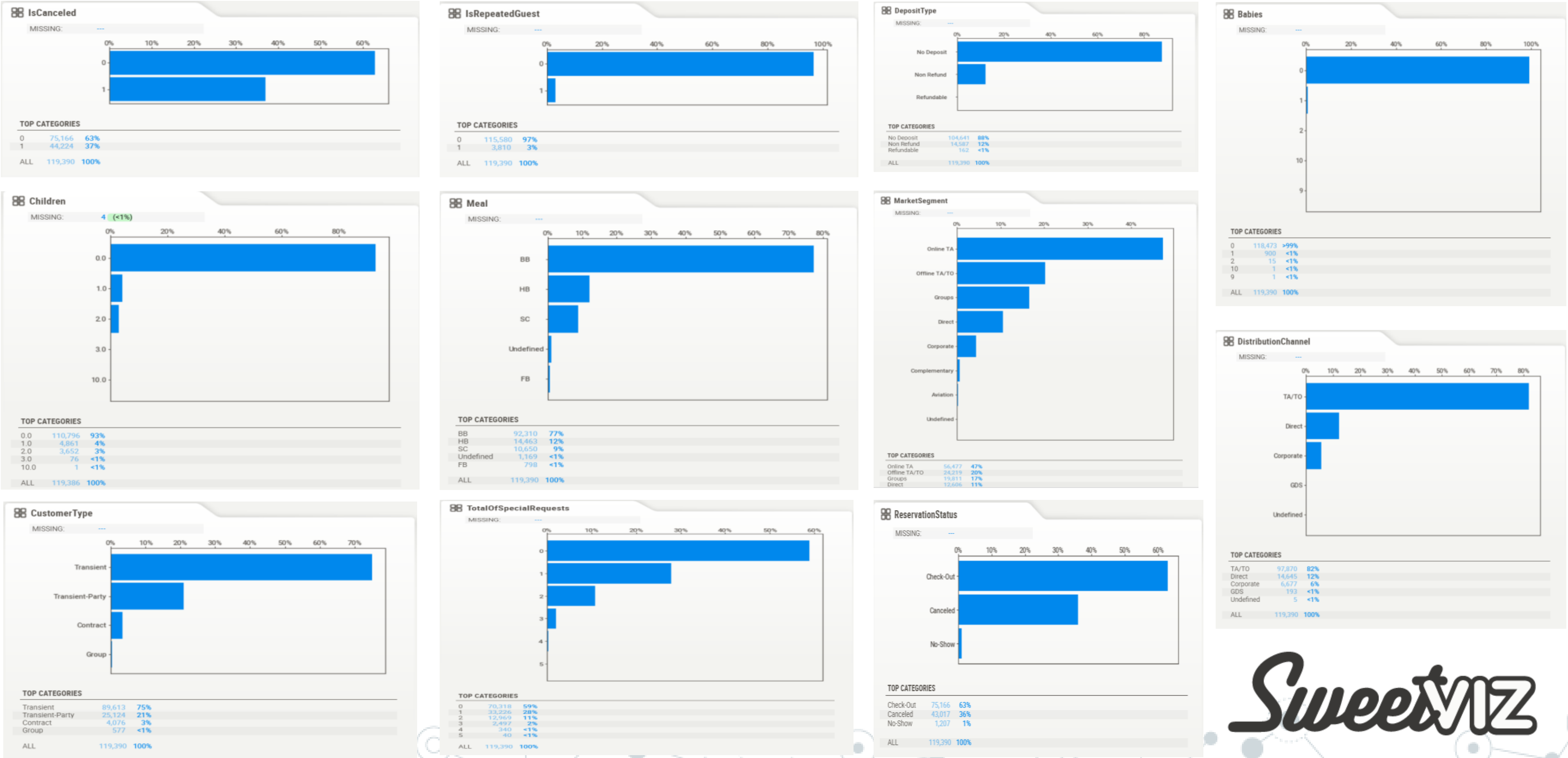
1) H1.csv – 40,060 rows
2) H2.csv – 79,310 rows

Split:

80% training data and 20% test data

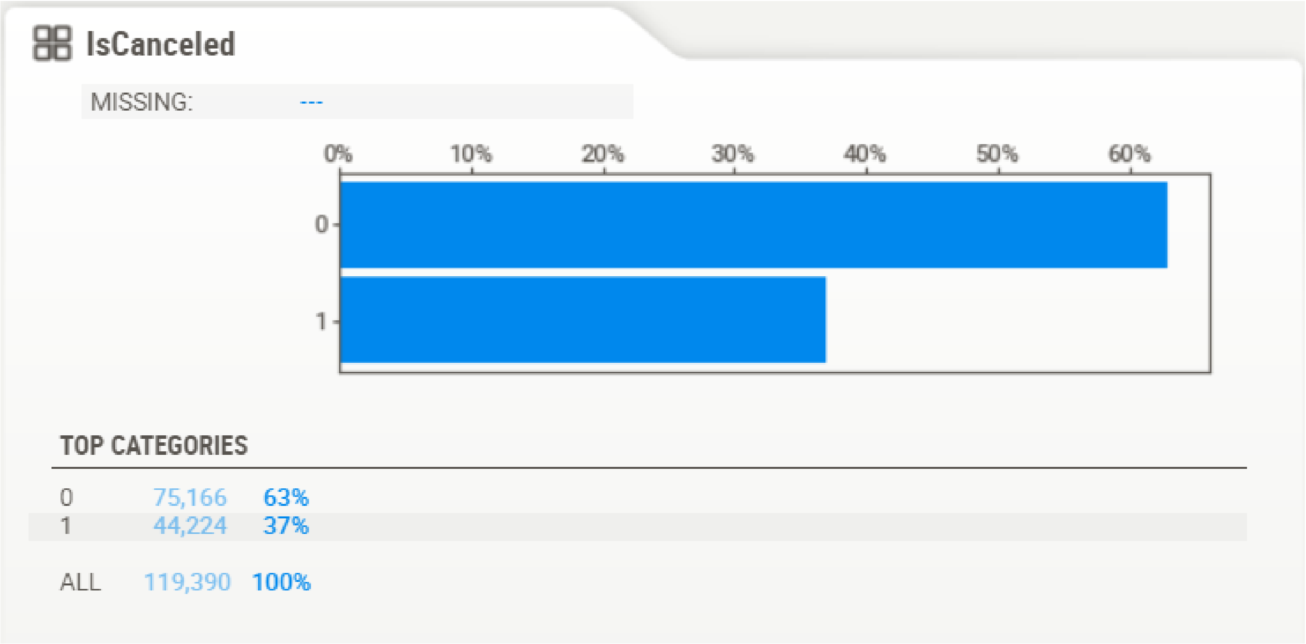
EDA

We used **SweetViz** library to aid in visualizing our overall dataset.

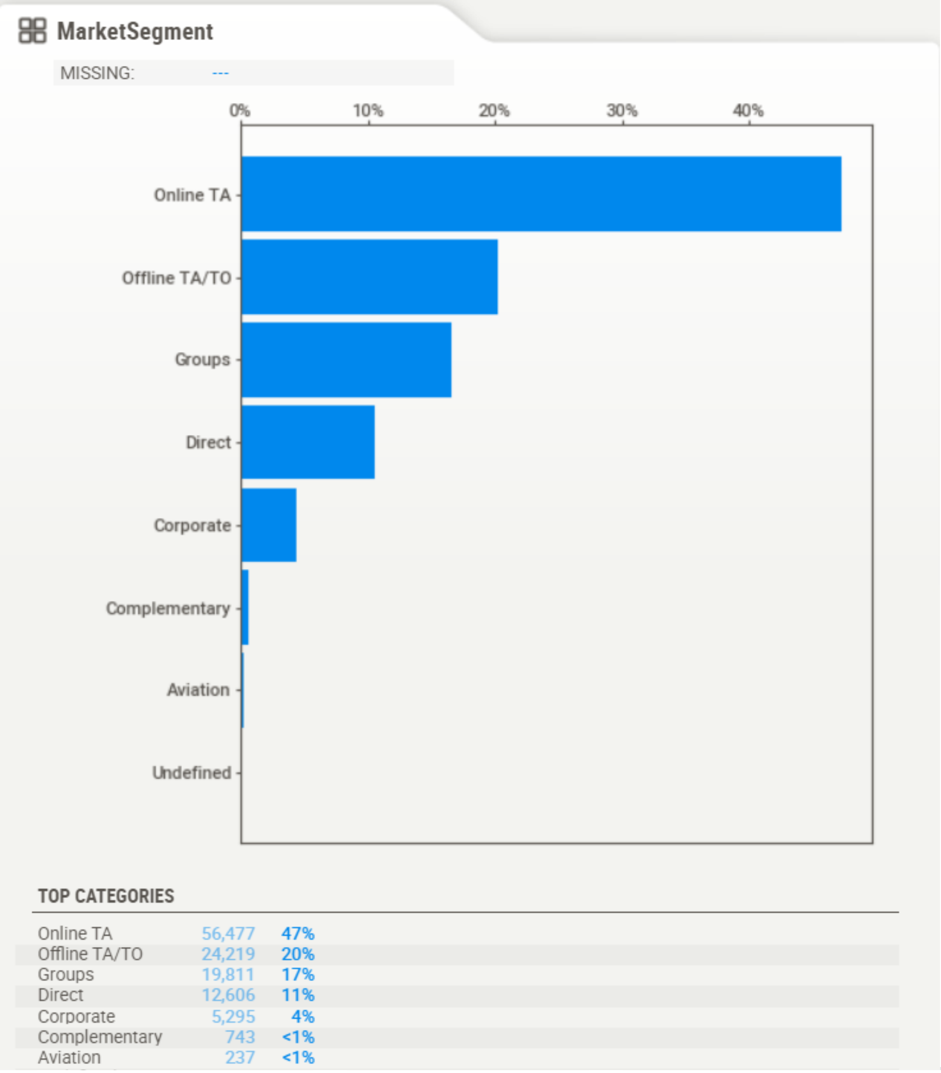


EDA

63% of bookings were not cancelled, **37%** of the bookings were cancelled



EDA



Bookings made via:

Online Travel Agent – 47%

Offline Travel Agents – 20%

Groups – 17%

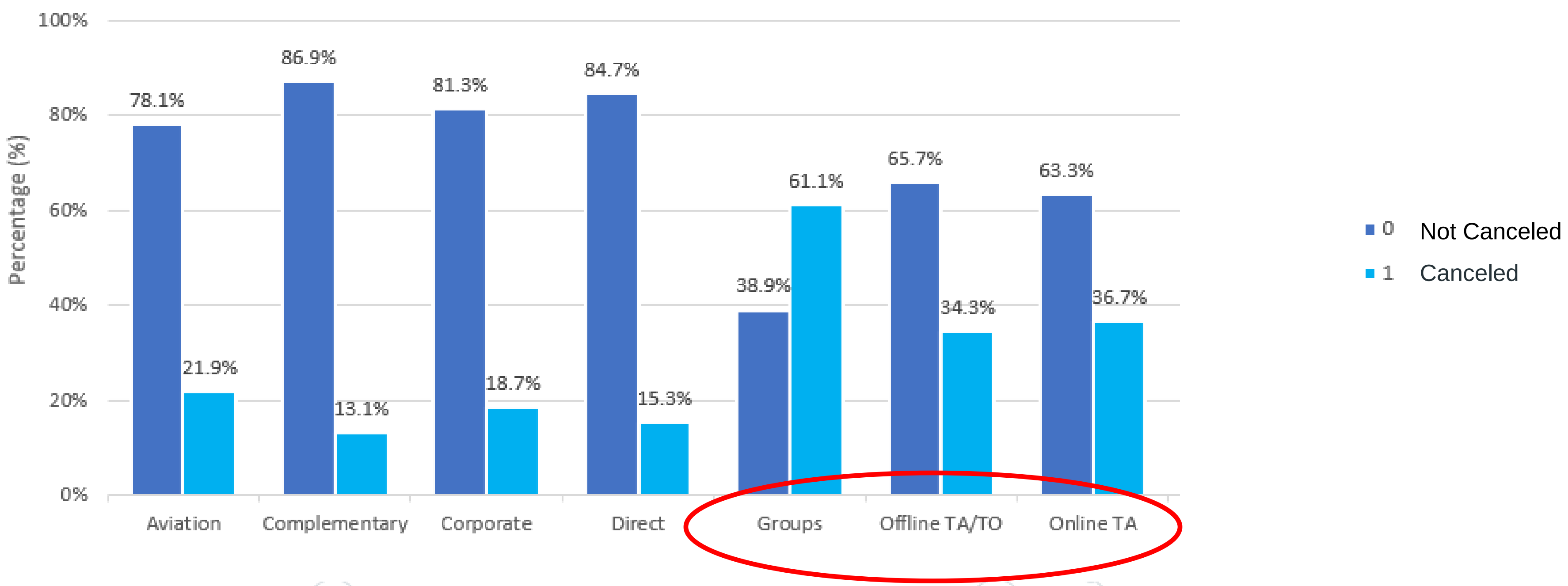
Direct – 11%

Others – < 5%

SweetVIZ

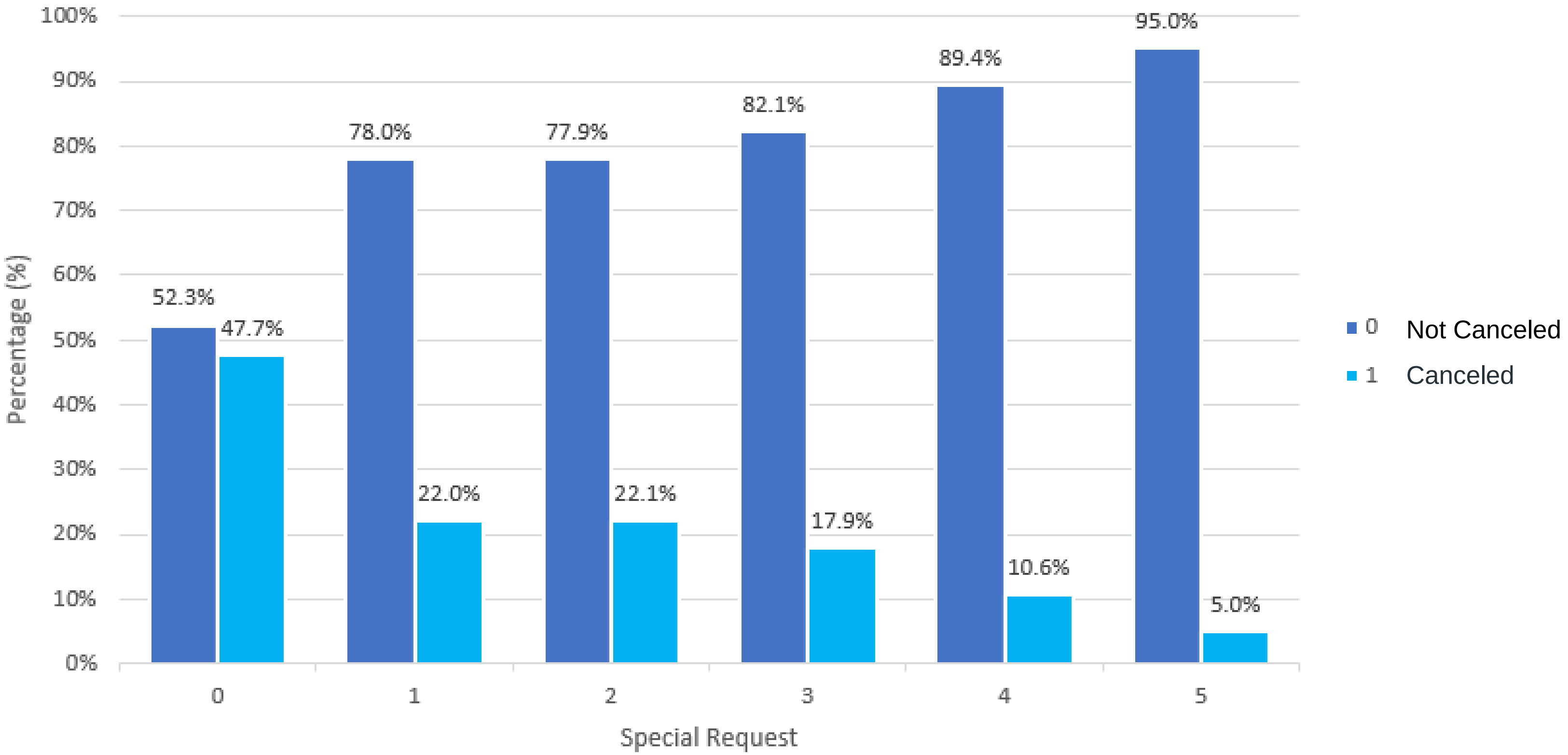
EDA

Cancelling rate for the top 3 Market Segment is the highest



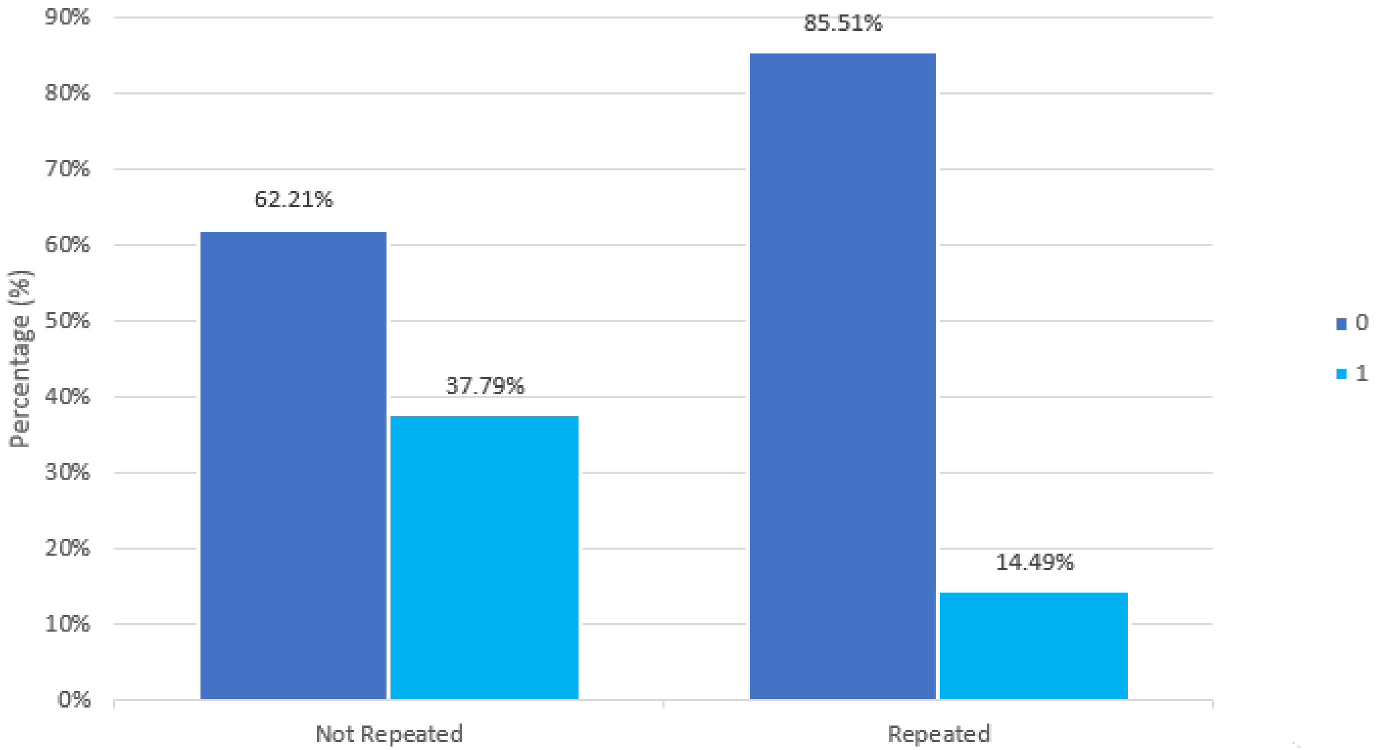
EDA

Inverse relationship between Number of Special Request made and cancelling rate



EDA

Returned customers lower tendency to cancel



Not Canceled
Canceled

Feature Engineering for All Model

Based on SweetViz charting, we performed feature engineering as follow:

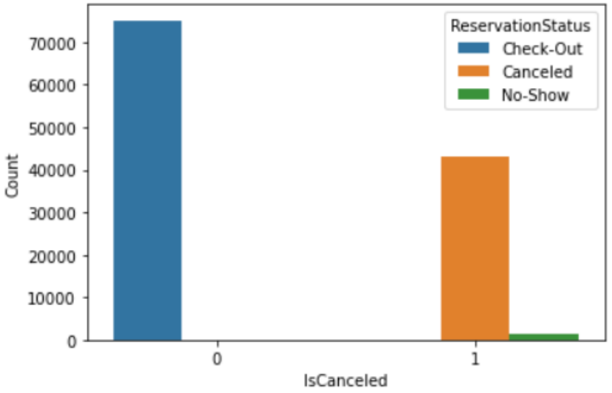
- Drop features that is not meaningful to the target
  - 'ArrivalDateYear', 'ArrivalDateDayOfMonth'

17	4406
5	4317
15	4196
25	4160
26	4147
9	4096
12	4087
16	4078
2	4055
19	4052
20	4032
18	4002
24	3993
28	3946
8	3921

3	3855
30	3853
6	3833
14	3819
27	3802
21	3767
4	3763
13	3745
7	3665
1	3626
23	3616
11	3599
22	3596
29	3580
10	3575
31	2208

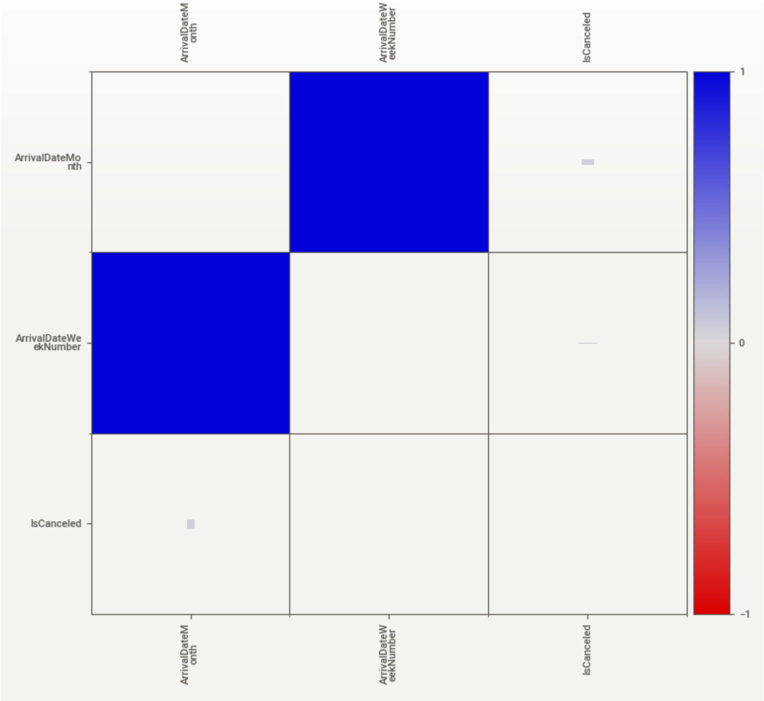
- Quasi Separation Issue
  - ‘ReservationStatusDate’, ‘ReservationStatus’, ‘RequiredCarParkingSpaces’

		Count
IsCanceled	**ReservationStatus**	
0	**Check-Out**	75166
1	**Canceled**	43017
	No-Show	1207

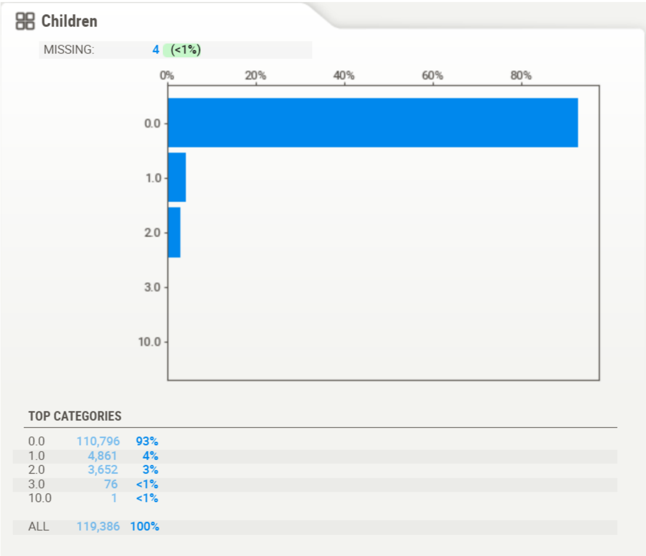


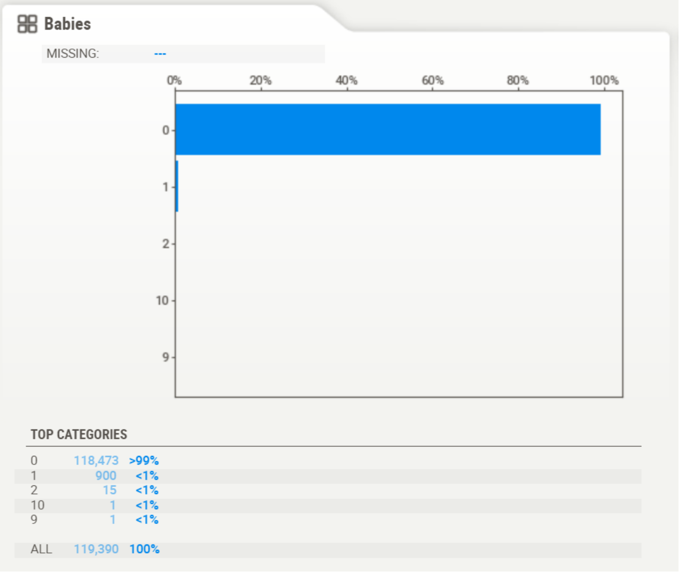
Feature Engineering for All Model

- Highly Correlation Feature
  - ‘ArrivalDateMonth’



- Outlier
  - ‘ADR’, ‘Adults’, ‘StaysInWeekNights’, ‘Babies’, ‘Children’





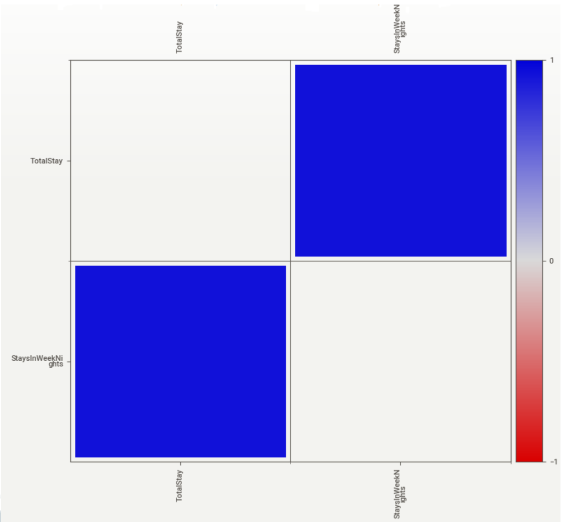
Additional Feature Engineering for Naïve Bayes

- New Column Added
  - ‘Region’, ‘TotalStay’, ‘SameRoomAssigned’
- Highly Correlation Feature
  - ‘StaysInWeekNights’

```
Europe countries:
array(['GBR', 'PRT', 'BEL', 'DEU', 'IRL', 'RUS', 'ESP', 'AUT', 'NLD',
       'FRA', 'ITA', 'LUX', 'FIN', 'POL', 'CHE', 'DNK', 'NOR', 'ROU',
       'SWE', 'HUN', 'HRV', 'JEY', 'LVA', 'SVN', 'UKR', 'SRB', 'MCO',
       'CZE', 'BGR', 'EST', 'GRC', 'ALB', 'SVK', 'BIH', 'BLR', 'LTU',
       'MNE', 'ISL', 'AND', 'MLT', 'GIB', 'LIE', 'MKD', 'FRO', 'IMN',
       'GGY', 'SMR'], dtype=object)
```

	AssignedRoomType	ReservedRoomType	SameRoomAssigned
0	A	A	1
1	A	A	1
2	D	D	1
3	A	A	1
4	D	A	0
5	A	A	1

	TotalStay	StaysInWeekendNights	StaysInWeekNights
0	4	2	2
1	3	2	1
2	5	2	3
3	3	0	3
4	3	1	2
...	...	...	...
92431	1	0	1
92432	4	1	3
92433	5	0	5
92434	3	0	3
92435	10	3	7



Additional Feature Engineering for Naïve Bayes

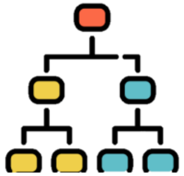
- Aggregated Representative:
  - ‘Country’
- Convert Features into Binary Value
  - Features: ‘Hotel’, 'Children', 'Babies', 'PreviousCancellations', 'PreviousBookingsNotCanceled', 'BookingChanges', 'Agent', 'Company', 'DaysInWaitingList'
  - For all 0 and NaN value -> 0
  - All other than above -> 1

Feature Engineering

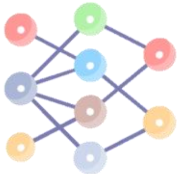
- One-Hot-Encoding on categorical variables:
  - Generic Model: All categorical features
  - Naïve Bayes Model: All categorical features that are non-binary value
- Scaling:
  - StandardScaler: Logistic Regression, K-Nearest Neighbor, Neural Network
  - MinMaxScaler: Naïve Bayes (except Gaussian Naïve Bayes)

Models

Random Forest

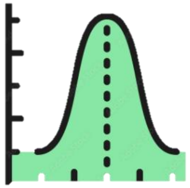


Neural Network



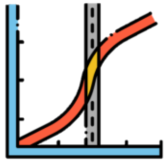
KNN

Classification Models



Naïve Bayes

Logistic Regression



Naïve Bayes

Since Naïve Bayes do not have hyperparameter besides alpha, we also tried multiple variations of Naïve Bayes model and compared their F1 Score.

As alpha increases, the likelihood probability moves toward uniform distribution

Alpha does not change the performance of the model that much

Naïve Bayes Model Used:

- Gaussian Naïve Bayes
- Categorical Naïve Bayes
- Multinomial Naïve Bayes
- Complement Naïve Bayes
- Gaussian Naïve Bayes + Categorical Naïve Bayes

Model (common feature engineering)	Test Accuracy
GaussianNB	0.464301168
CategoricalNB	0.75049762
MultinomialNB	0.749675465
ComplementNB	0.747468628
GaussianNB + CatNB	0.762786672

Naïve Bayes

We noticed an extremely low accuracy on GaussianNB model, this might be due to

- Naïve Bayes performs better with categorical variables
- Sensitive to outliers and one-sided numerical data as they will affect the mean and standard deviation
- Gaussian perform internal standardization by calculating based on distance from center of distribution

Further feature engineering specifically for Naïve Bayes model.

Model (NB feature engineering)	Test Accuracy	Cross Validation mean (std)
GaussianNB	0.624275206	0.627 (0.007)
CategoricalNB	0.712159238	0.717 (0.002)
MultinomialNB	0.750930333	0.752 (0.001)
ComplementNB	0.703331891	0.704 (0.002)
GaussianNB + CatNB	0.750367806	-

Random Forest

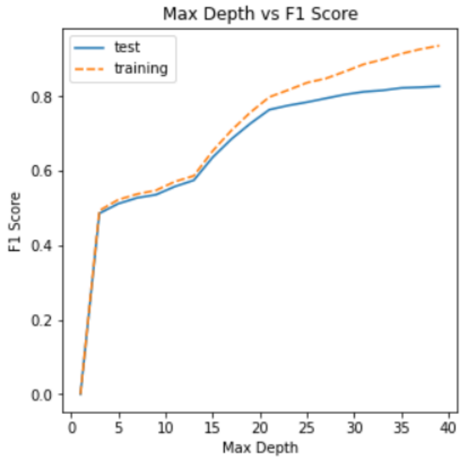
For random forest hyper-parameter tuning, we tried two–steps approach using GridSearchCV and plotting the numerical hyper-parameter with F1 Score.

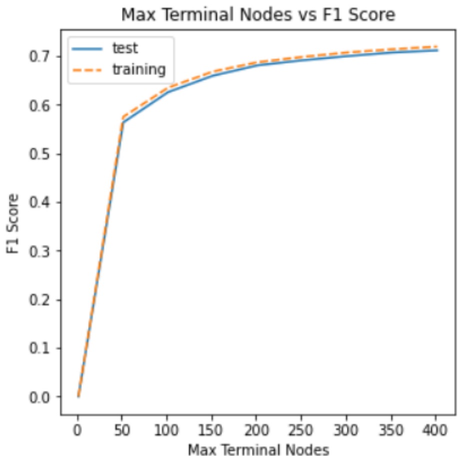
Step 1: Optimize the following three parameters

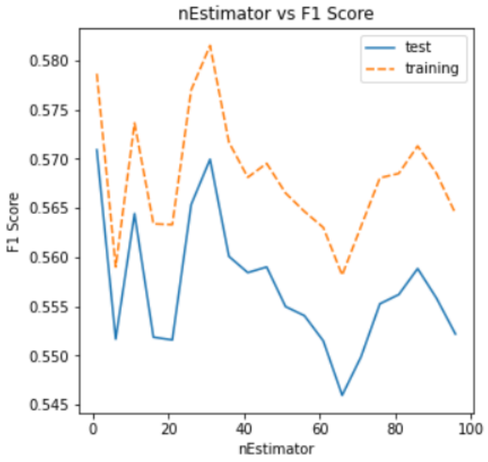
- max_features: ['auto', 'sqrt', 'log2'],
- oob_score: [True, False],
- criterion :['gini', 'entropy']

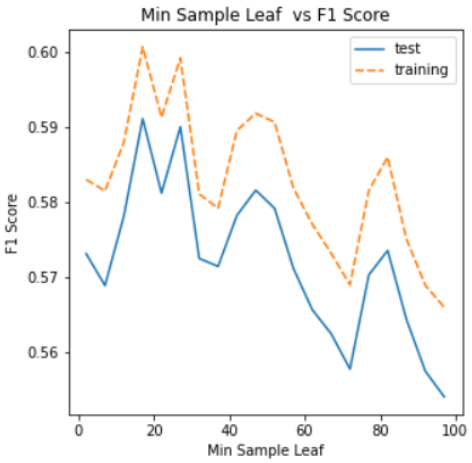
```
Chosen parameters:

random_state= 2021, max_features='auto', criterion="gini“,
oob_score=True, max_depth=20,
min_samples_split=2,max_leaf_nodes=40,n_estimators= 30,
min_samples_leaf=20
```

Step 2: Optimize the following numerical features:



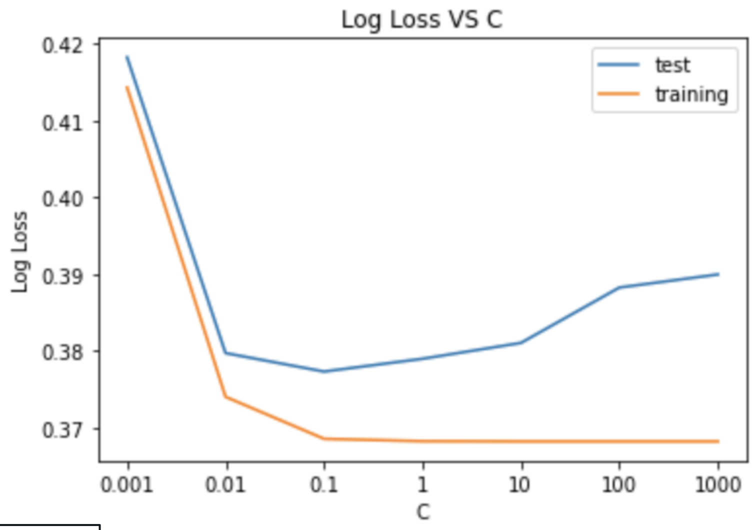






Logistic Regression

For logistic regression hyper-parameter tuning, we tried two–steps approach using GridSearchCV and plotting the numerical hyper-parameter with F1 Score.



Step 1: Optimize the following two parameters

- penalty: [**'l1',** 'l2’, ‘elasticnet’, ‘none’],
- solver: [**'liblinear’**, ‘sag’, ‘saga’, 'newton-cg’, 'lbfgs’]

Step 2: Optimize regularization C

- C: [0.001, 0.01, **0.1**, 1, 10, 100, 1000]

Without hyper parameter tuning

```
Confusion_matrix:
[[12973  1438]
 [ 2742  5957]]
Accuracy: 0.8191259195153613
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.90      0.86     14411
           1       0.81      0.68      0.74      8699

    accuracy                           0.82     23110
   macro avg       0.82      0.79      0.80     23110
weighted avg       0.82      0.82      0.82     23110
```

With hyper parameter tuning

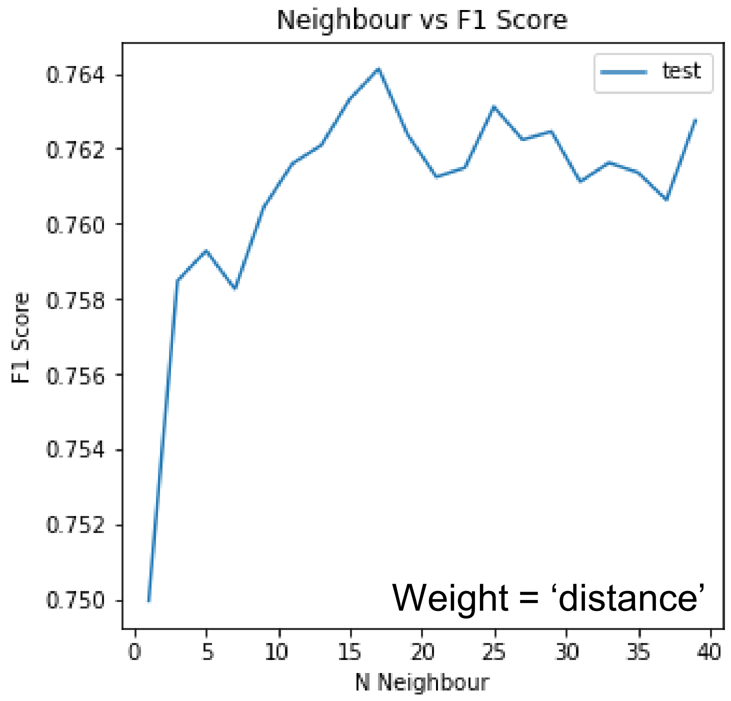
```
Confusion_matrix:
[[12985  1426]
 [ 2755  5944]]
Accuracy: 0.8190826482042406
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.90      0.86     14411
           1       0.81      0.68      0.74      8699

    accuracy                           0.82     23110
   macro avg       0.82      0.79      0.80     23110
weighted avg       0.82      0.82      0.82     23110
```

K Nearest Neighbour

- Used standard scaler

- Hyperparameter tuning:
  - weights: [‘uniform’, **‘distance’**],
  - n_neighbors: odd numbers in between 1 – 40

  → Best F1 score at n_neighbors = **17**



Weight = ‘uniform’, n_neighbors = 1

```
Confusion_matrix:
[[12041  2485]
 [ 2052  6814]]
    Accuracy: 0.8060448016415869
    Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.83      0.84     14526
           1       0.73      0.77      0.75      8866

    accuracy                           0.81     23392
   macro avg       0.79      0.80      0.80     23392
weighted avg       0.81      0.81      0.81     23392
```

Weight = ‘distance’, n_neighbors = 17

```
Confusion_matrix:
[[12762  1649]
 [ 2301  6398]]
Accuracy: 0.8290783210731285
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87     14411
           1       0.80      0.74      0.76      8699

    accuracy                           0.83     23110
   macro avg       0.82      0.81      0.82     23110
weighted avg       0.83      0.83      0.83     23110
```

Neural Network

- Tried the following hyperparameter tuning:

✓ Learning rate = [**0.01**, 0.02, 0.03],
✓ Batch size = [16, 32, 64, 128, 256, 512, **1024**]
✓ Optimizer = [**Adam**, SGD]
✓ Epoch = [20, 50, **100**]

- Used Standard Scaler

- Tried with/without down sampling

```
# example: batch size = 16 with 2 hidden layers
model = Sequential()
model.add(Dense(50, input_dim=x_train_one_hot_data.shape[1], activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(20, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(10, activation='relu'))
model.add(Dropout(0.2))
model.add(layers.Dense(1, activation='sigmoid'))
```

	precision	recall	f1-score
0	0.87	0.92	0.89
1	0.84	0.76	0.80
accuracy			0.86
macro avg	0.86	0.84	0.85
weighted avg	0.86	0.86	0.86

```
# Best results: batch size = 1024 with 3 hidden layers, without down sampling
modelFinal = Sequential()
modelFinal.add(Dense(78, input_dim=x_train_one_hot_data.shape[1], activation='relu'))
modelFinal.add(Dropout(0.2))
modelFinal.add(Dense(39, activation='relu'))
modelFinal.add(Dropout(0.2))
modelFinal.add(Dense(19, activation='relu'))
modelFinal.add(Dropout(0.2))
modelFinal.add(Dense(10, activation='relu'))
modelFinal.add(Dropout(0.2))
modelFinal.add(layers.Dense(1, activation='sigmoid'))
```

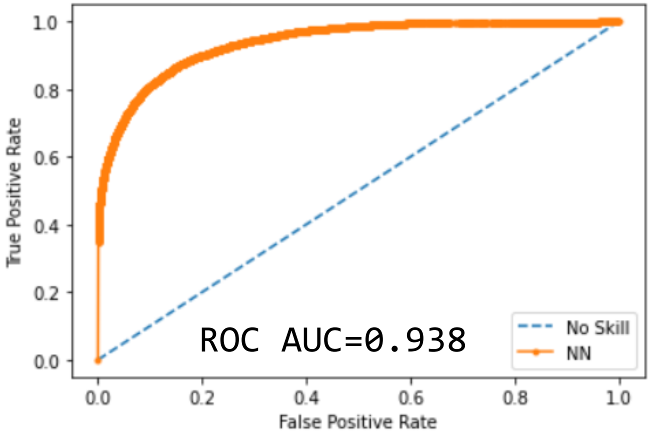
	precision	recall	f1-score	support
0	0.87	0.92	0.89	14411
1	0.85	0.78	0.81	8699
accuracy			0.86	23110
macro avg	0.86	0.85	0.85	23110
weighted avg	0.86	0.86	0.86	23110

Model Results Comparison

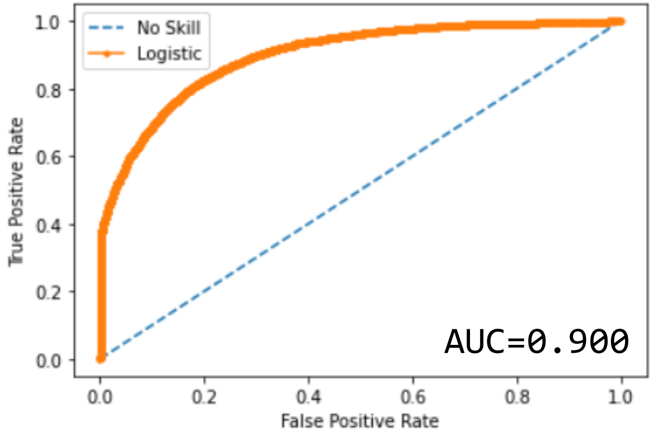
Machine Learning Models	Accuracy	Precision	Recall	F1	Log Loss
Neural Network	0.86	0.85	0.78	0.81	4.67
Random Forest	0.83	0.82	0.71	0.76	5.87
K Nearest Neighbors	0.83	0.80	0.74	0.76	5.90
Logistic Regression	0.82	0.81	0.68	0.74	6.25
Multinomial NB	0.75	0.75	0.51	0.61	8.60

ROC Comparison

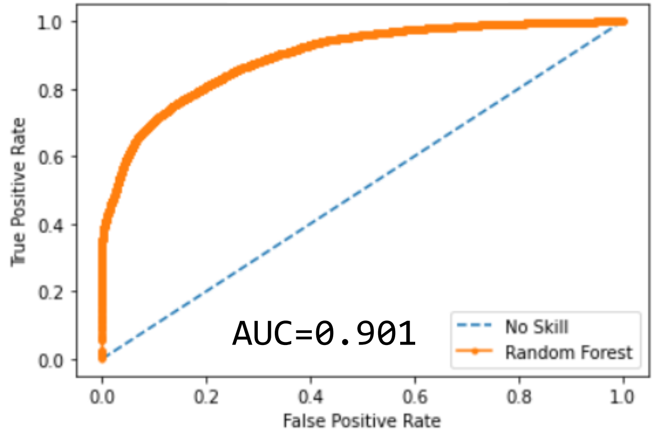
Neural Network



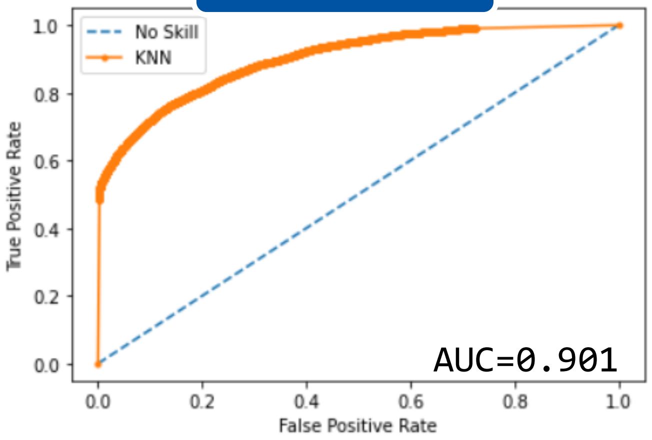
Logistic Regression



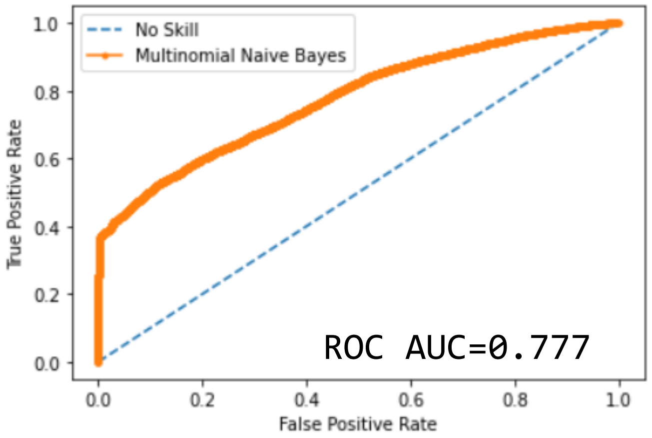
Random Forest



KNN



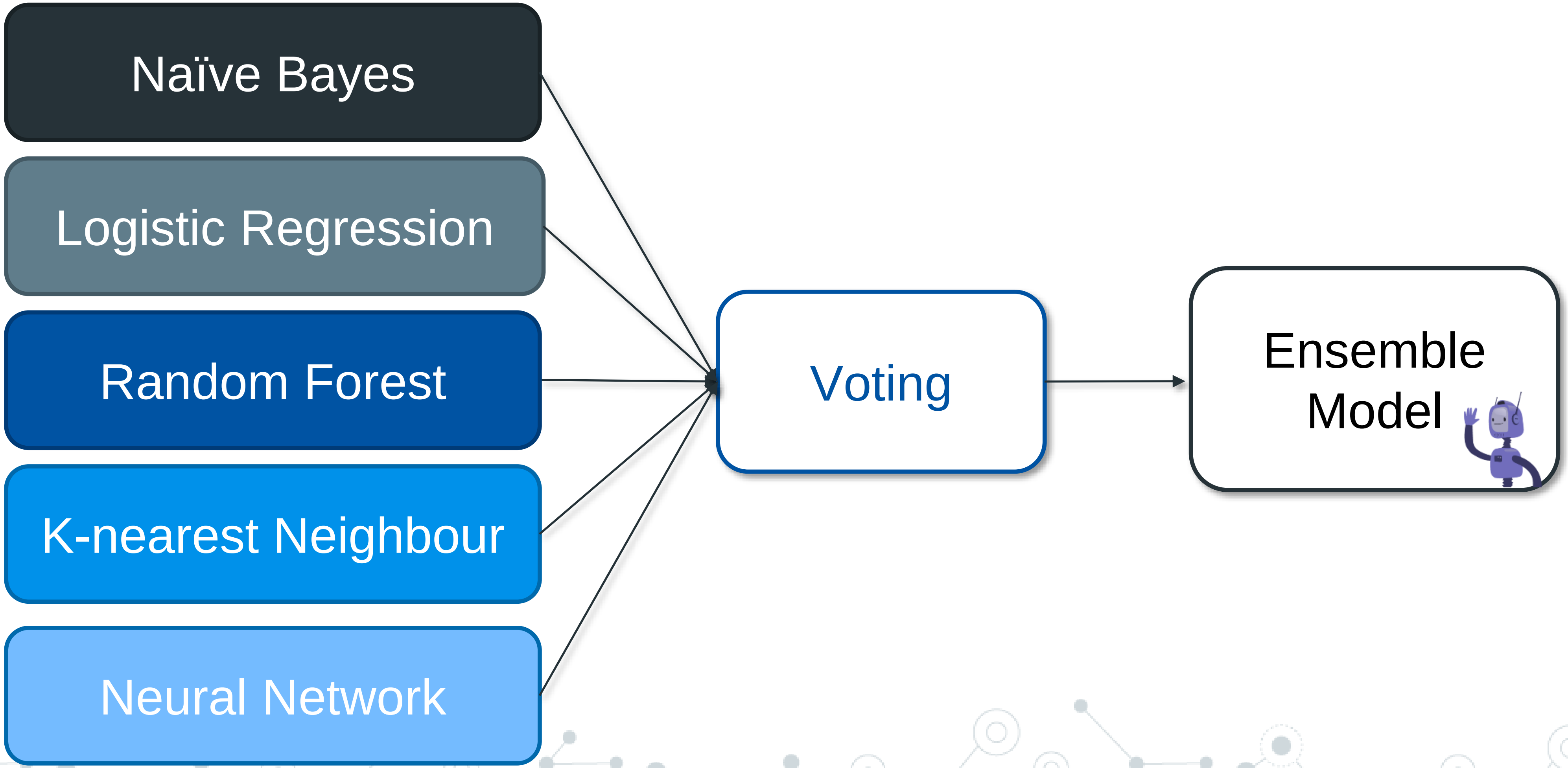
Naïve Bayes



Model Specification Comparison

Machine Learning Models	Model Size	Training Time	Test Time	Scaling
Neural Network	1.05 MB	52s	1s	Yes
Random Forest	1.23 MB	7s	1s	No
K Nearest Neighbors	667 MB	56s	46s	Yes
Logistic Regression	7 kb	49s	1s	Yes
Multinomial NB	4 kb	3s	1s	Yes

Ensemble – Majority Voting



Ensemble – Majority Voting

Machine Learning Models	Accuracy	Precision	Recall	F1
Neural Network	0.86	0.85	0.78	0.81
Random Forest	0.83	0.82	0.71	0.76
K Nearest Neighbors	0.83	0.80	0.74	0.76
Logistic Regression	0.82	0.81	0.68	0.74
Multinomial NB	0.75	0.75	0.51	0.61
Ensemble	0.85	0.88	0.70	0.78

Ensemble – Majority Voting Error Correlation

Voting Error Correlation

Ensemble – Majority Voting

Machine Learning Models	**Accuracy**	**Precision**	**Recall**	**F1**
Neural Network	0.86	0.85	0.78	0.81
Random Forest	0.83	0.82	0.71	0.76
K Nearest Neighbors	0.83	0.80	0.74	0.76
Logistic Regression	0.82	0.81	0.68	0.74
Multinomial NB	0.75	0.75	0.51	0.61
Ensemble	0.85	0.88	0.70	0.78
Ensemble - After Dropping Random Forest and Logistic Regression Models	0.86	0.89	0.71	0.79

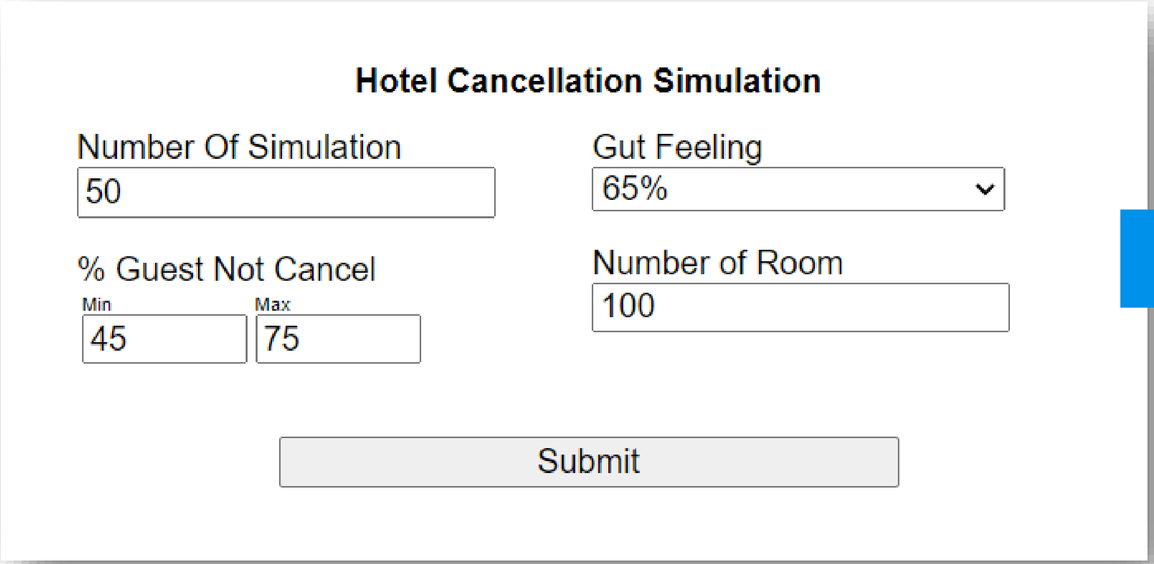
Implication for Business

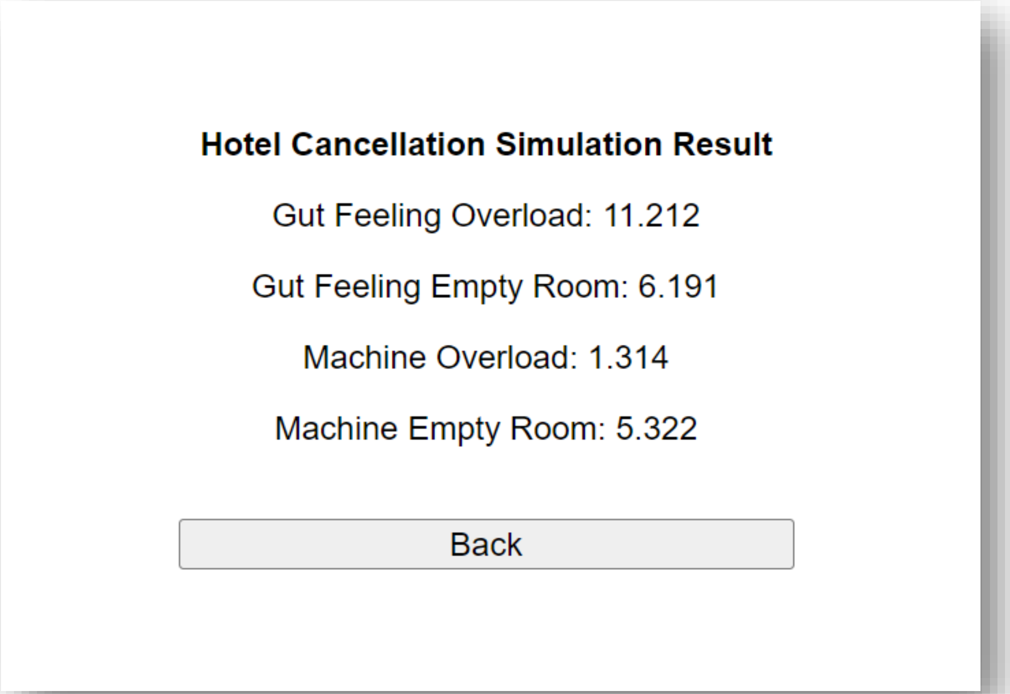
1000 simulation run for scenario that expected cancellation rate changes in the future to vary in the range of 35% to 45% randomly

- Gut Feeling : Cancellation rate ~40%, based on historical cancellation rate
- Machine learning: Predict cancellation based on features
- Result:

For every 100 room	Gut Feeling	Machine Learning
no of room overbooked	11.21	1.31
no of room underbooked	6.19	5.32
Observation	Will result in the huge error; leading to potential revenue loss	Consistently low error for both overbooking and underbooking

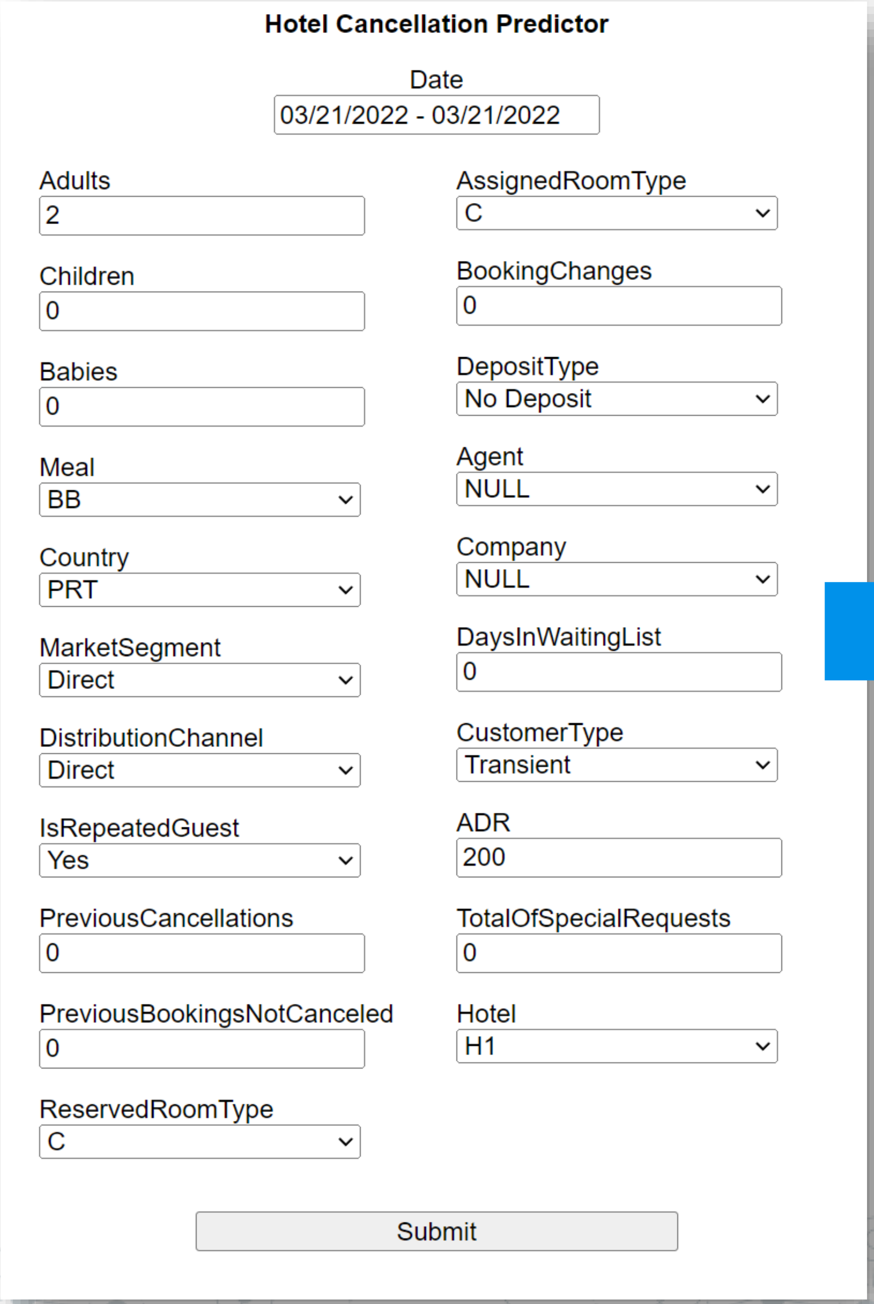
Simulation

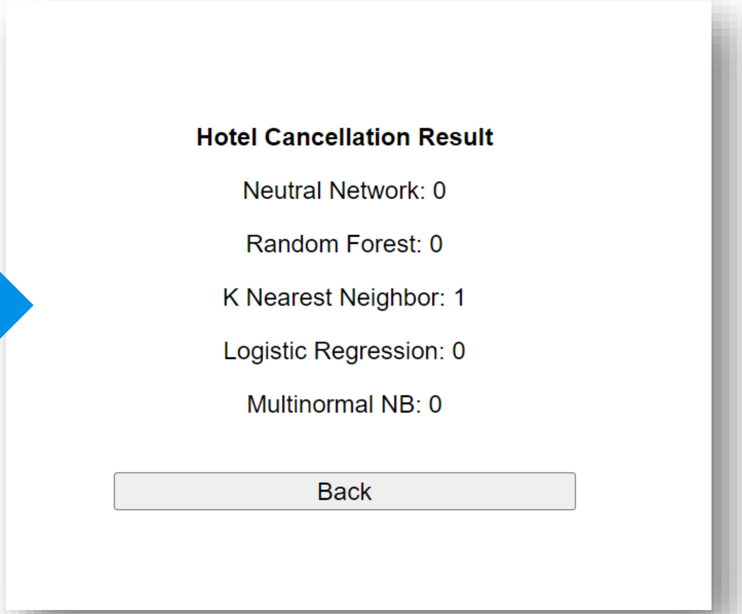




Flask

DEMO


Hotel Cancellation Predictor
Date
03/21/2022 - 03/21/2022
Adults
2
Children
0
Babies
0
Meal
BB
Country
PRT
MarketSegment
Direct
DistributionChannel
Direct
IsRepeatedGuest
Yes
PreviousCancellations
0
PreviousBookingsNotCanceled
0
ReservedRoomType
C
AssignedRoomType
C
BookingChanges
0
DepositType
No Deposit
Agent
NULL
Company
NULL
DaysInWaitingList
0
CustomerType
Transient
ADR
200
TotalOfSpecialRequests
0
Hotel
H1
Submit


Hotel Cancellation Result
Neutral Network: 0
Random Forest: 0
K Nearest Neighbor: 1
Logistic Regression: 0
Multinormal NB: 0
Back

Flask

Conclusion

- Neural network is the best machine learning model for the prediction,
- Ensemble learning does not guarantee better result as the weaker model drag down the overall ensemble model performance,
- Deploying the Machine Learning model will minimize the hotel room overbooking and maximize the hotel booking capacity
  - Improve customers' experience
  - Improve the hotel profit in the long run

Thank you! ☺