Predicting profit and Driving Business Insights using Iowa spirits sales

Author: Razvan Nelepcu

Mentor: Dhiraj Khanna

Project organization

The project follows all steps of Data Science workflow divided in the following major steps:

- Problem Identification
- Data wrangling
- Exploratory Data Analysis
- Pre-processing and training data development
- Modeling
- Documentation

Context

- The Covid-19 pandemic is the major event of the 3rd Millennium that changed lives and businesses.
- We know that some businesses went bankrupt or had to close their businesses overnight, such as bars, restaurant, hotels, or aviation. Some other flourished, such as online shopping, grocery stores or delivery services.
- And nowadays more than ever we need data to understand how the pandemic impacted different parts of economy.
- Our project will be directed on analyzing the spirits sales in Iowa since 2012 to present.

Problem statement

This project looks to resolve some issues that are of high importance for a diverse number of involved entities:

- Exploration on what was the impact of Covid-19 on the Alcoholic Beverages Industry.
- Storage capacity management exploratory analysis for Iowa Department of Commerce, Alcoholic Beverages Division.
- Cohort Analysis and Customer Segmentation using RFM(Recency, Frequency and Monetary value) and Unsupervised Learning
- Using time series analysis and predictions to predict profit for next month for Iowa Dept of Commerce from spirits.
- Lastly, we want to assist a hypothetical liquor store owner in Iowa in expanding to new locations throughout the state.

DATA COLLECTION AND ORGANIZATION

- The main dataset used is the Iowa Liquor Sales database from Data.Iowa.gov.
- It contains more than 24 million records of spirits purchase of Class "E" liquor licenses by product and date of purchase from January 1, 2012, to current, data provided and updated monthly by Iowa Department of Commerce, Alcoholic Beverages Division.
- The data contains labels such as Invoice number, Store, Address, Zip Code, Geographical Location, beverage category, vendor name, Item Description, State Bottle Cost, State Bottle Retail, Bottles Sold and Sale.

INITIAL NUMERIC DATA EXPLORATION

To clean data, we need to explore data.

We explored the numerical data distributions and outliers, and here are some findings:

- an average 37.04% cancellation rate for the 2 hotels
- that all the top 10 bookings with the highest number of adults were canceled
- A single outlier was identified as entry error and was removed from the dataset - a booking with an extremely high ADR value.

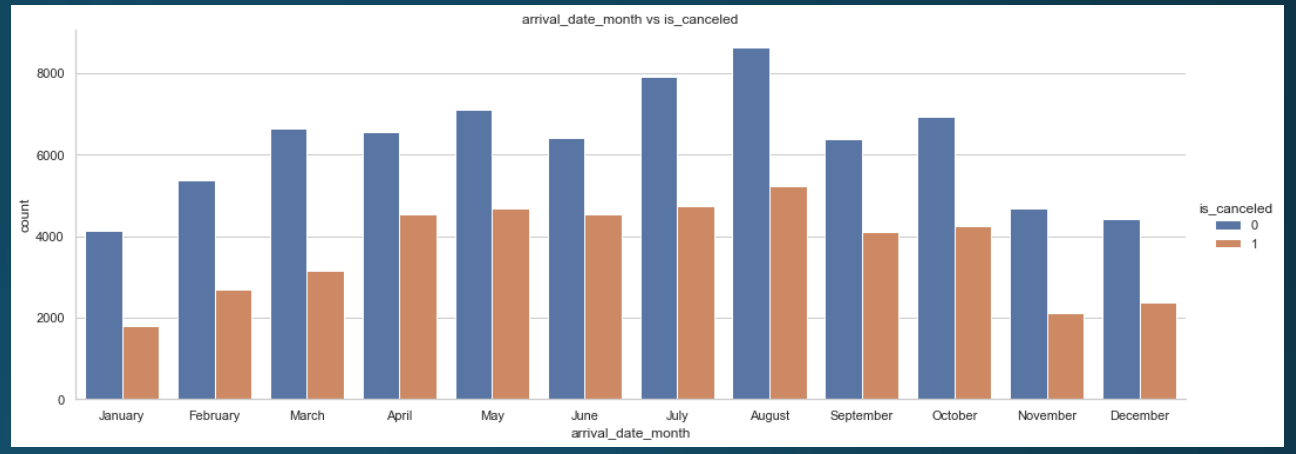
INITIAL CATEGORICAL DATA EXPLORATION

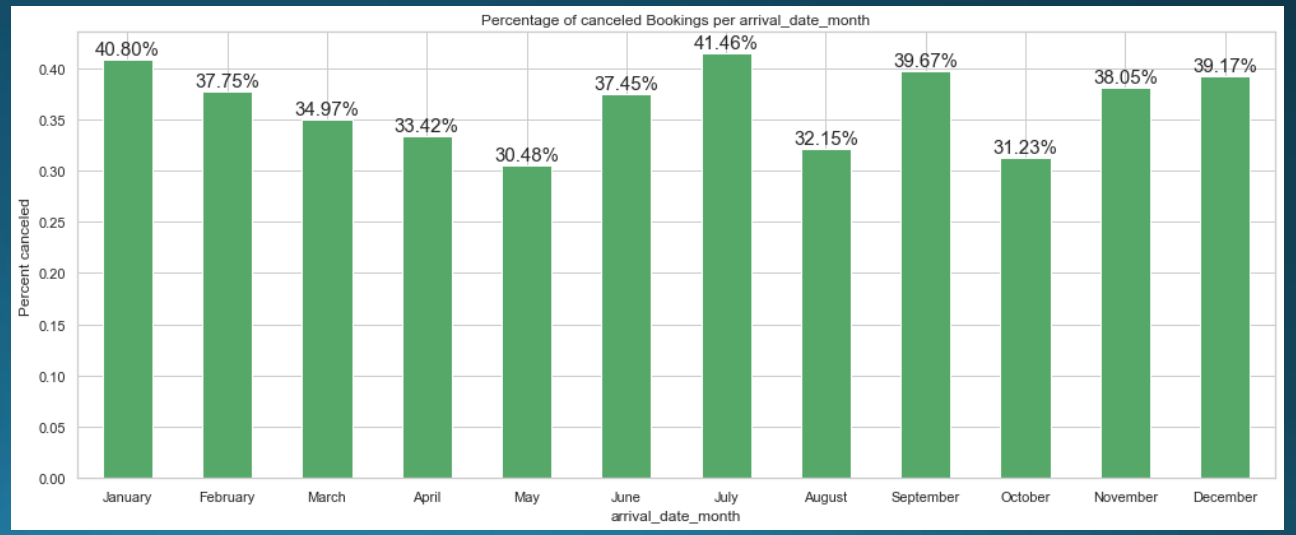
Out prediction target feature was the binomial feature **is_canceled.**

Conducted Data Visualizations of categorical features versus our target feature.

One interesting finding was the months distribution, and another was regarding the Deposit type feature

Completed by cleaning missing values and saving data.





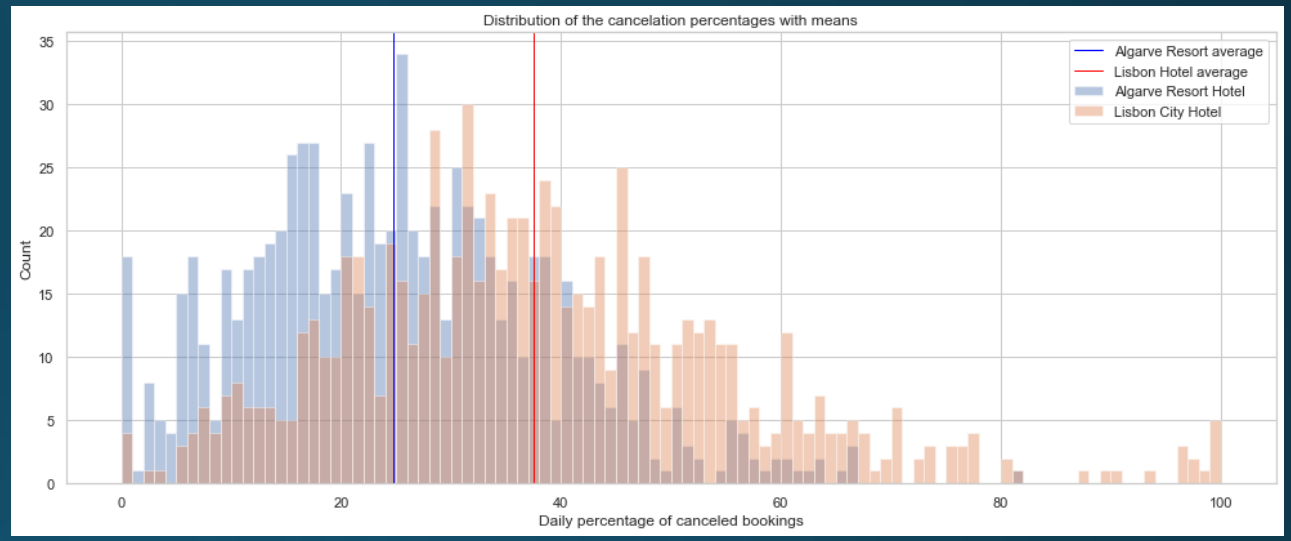
HYPOTHESIS TESTING

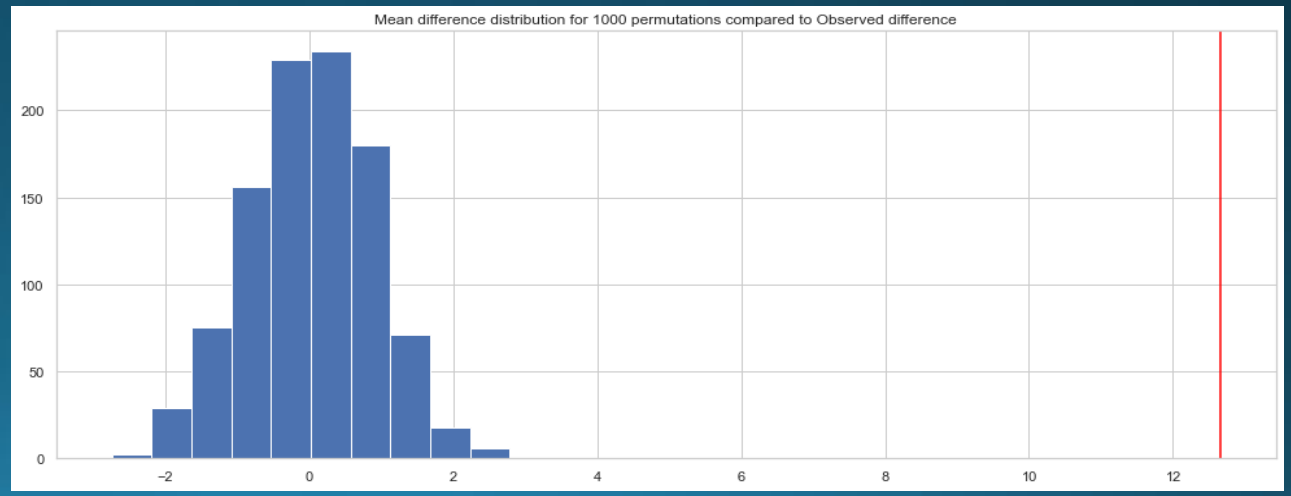
"EDA approach is precisely that - an approach - not a set of techniques, but an attitude/philosophy about how a data analysis should be carried out." **e-Handbook of Statistical Methods**

How different are the two hotel bookings cancellations?

We created the daily cancellations rates and used them to see the two hotels' distributions

Non-parametric test with 1000 permutations.



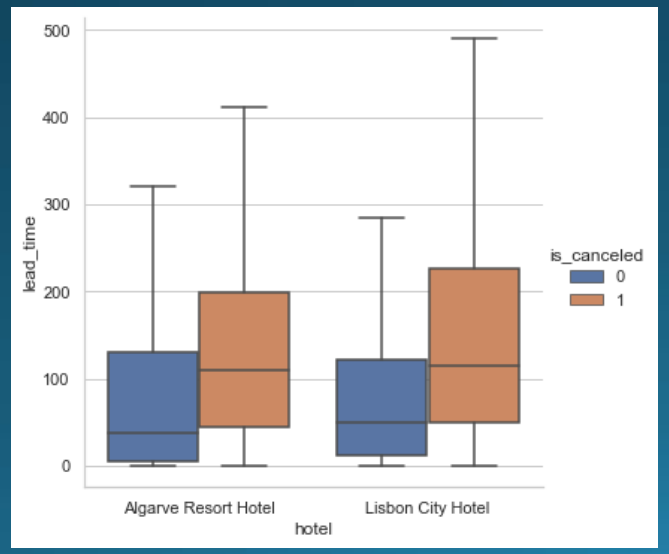


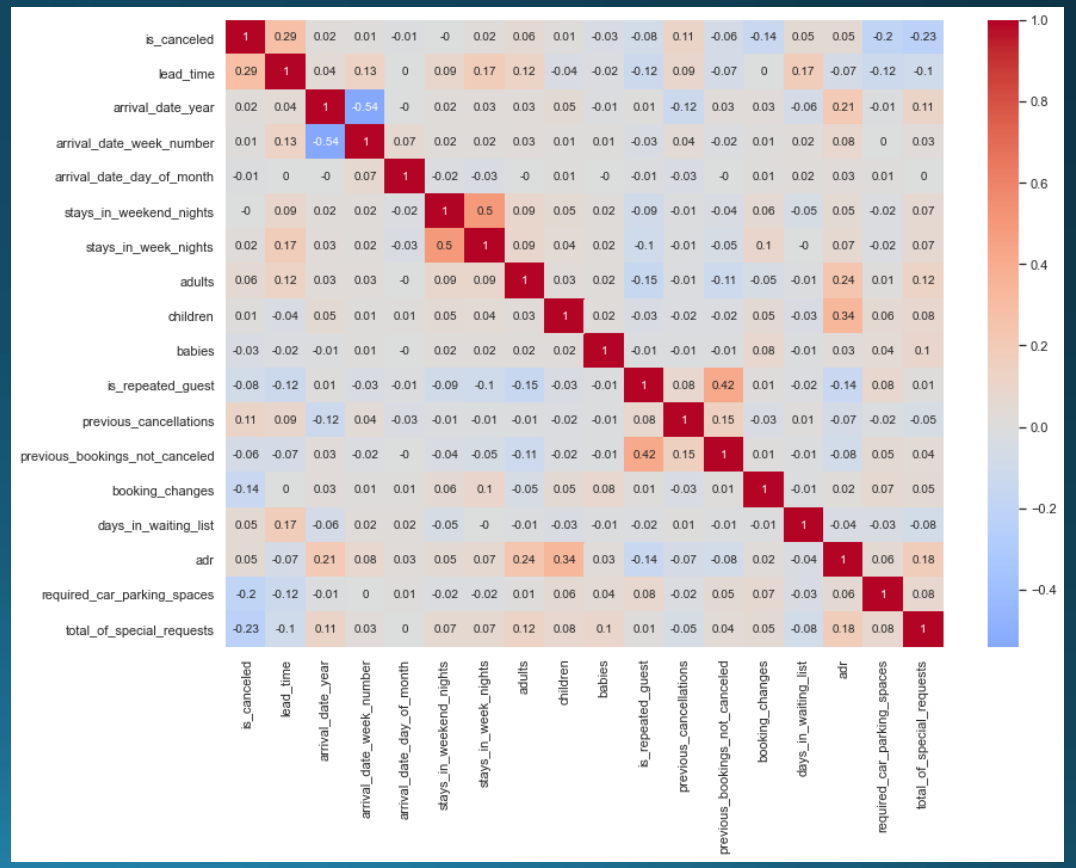
FEATURE ANALYSIS

Heatmap to visualize all correlations from numerical features

Starting from this we continued with bivariate EDA and multivariate EDA

High correlation between Is_canceled and lead_time.

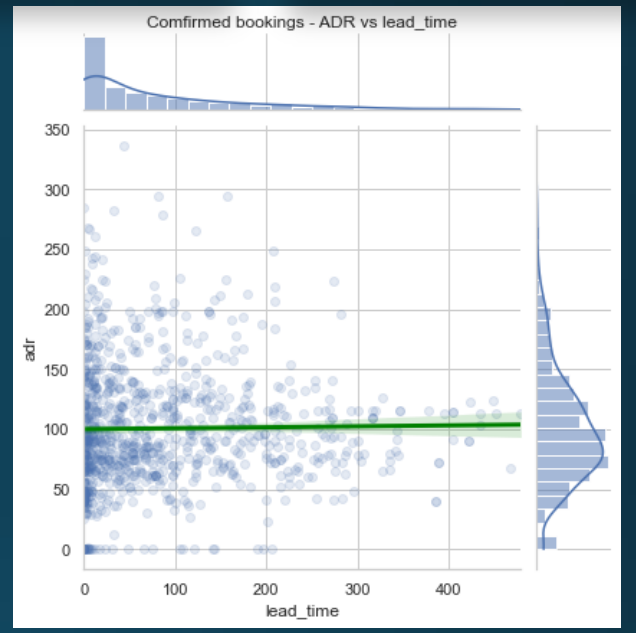


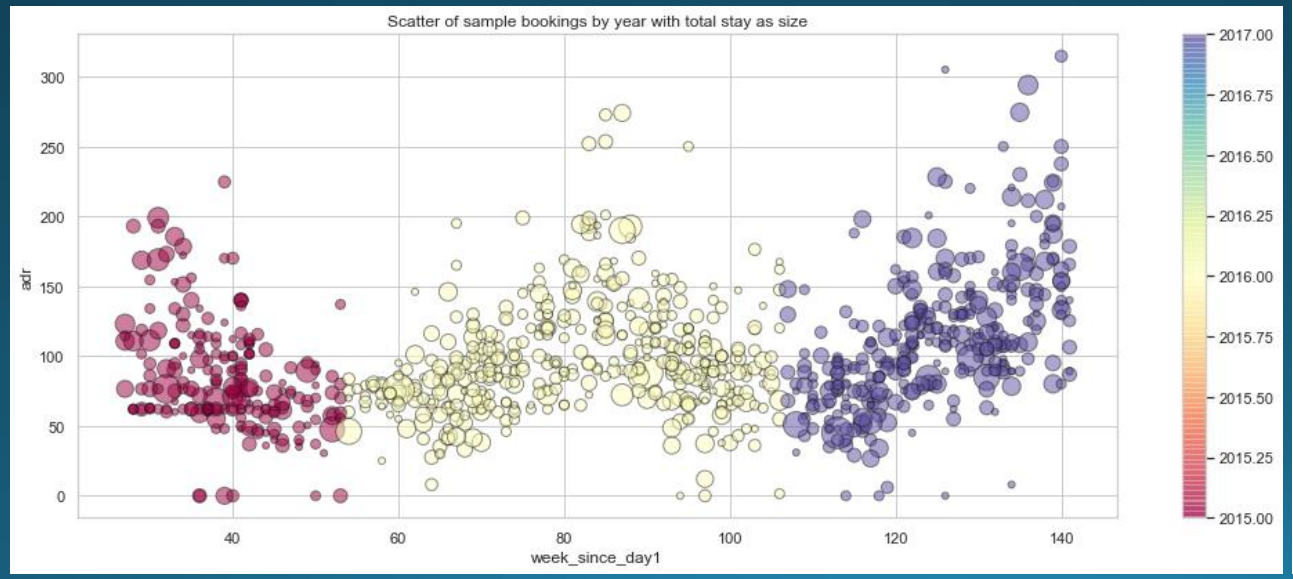


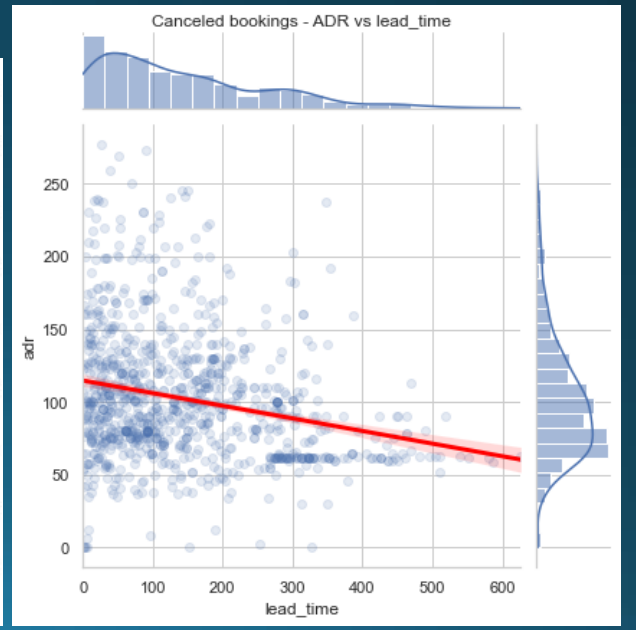
FEATURE ANALYSIS

Analyzed correlation between ADR(average daily rate) and lead_time on canceled and confirmed bookings subsets.

Plotted correlations between week number, ADR and length of stay.







NON-REFUNDABLE DEPOSIT

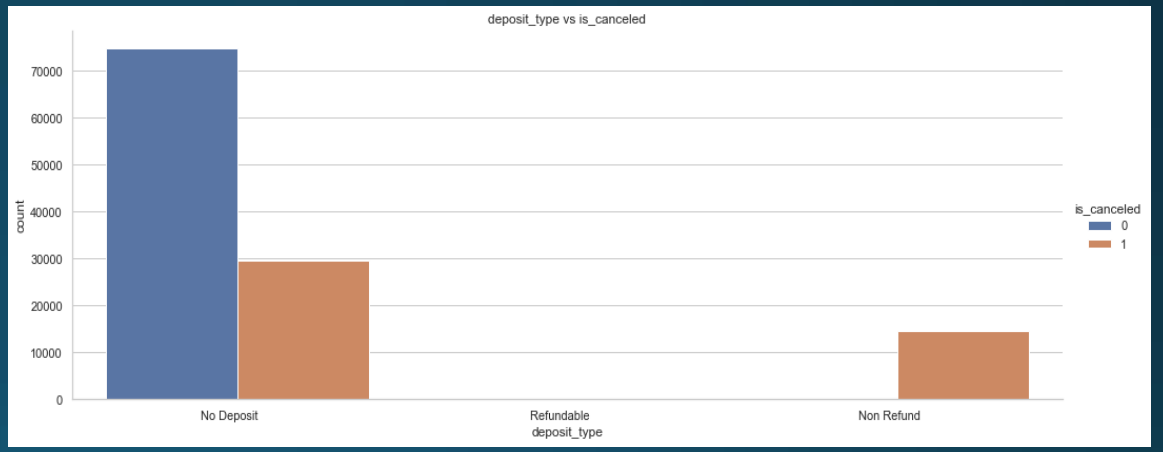
Issue identified in Data Wrangling section.

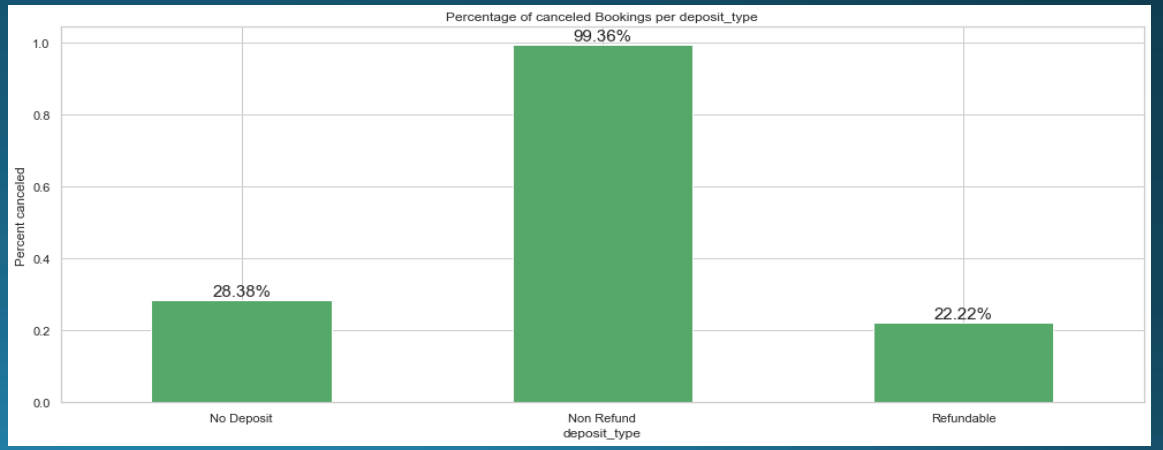
14,586 customers did a non-refundable deposit

“Non Refund – a deposit was made in the value of the total stay cost”- **Source Data Description**

Analyzed direct feature correlation

Analyzed the difference of correlations between the subset of non-refundable deposits and the whole data.





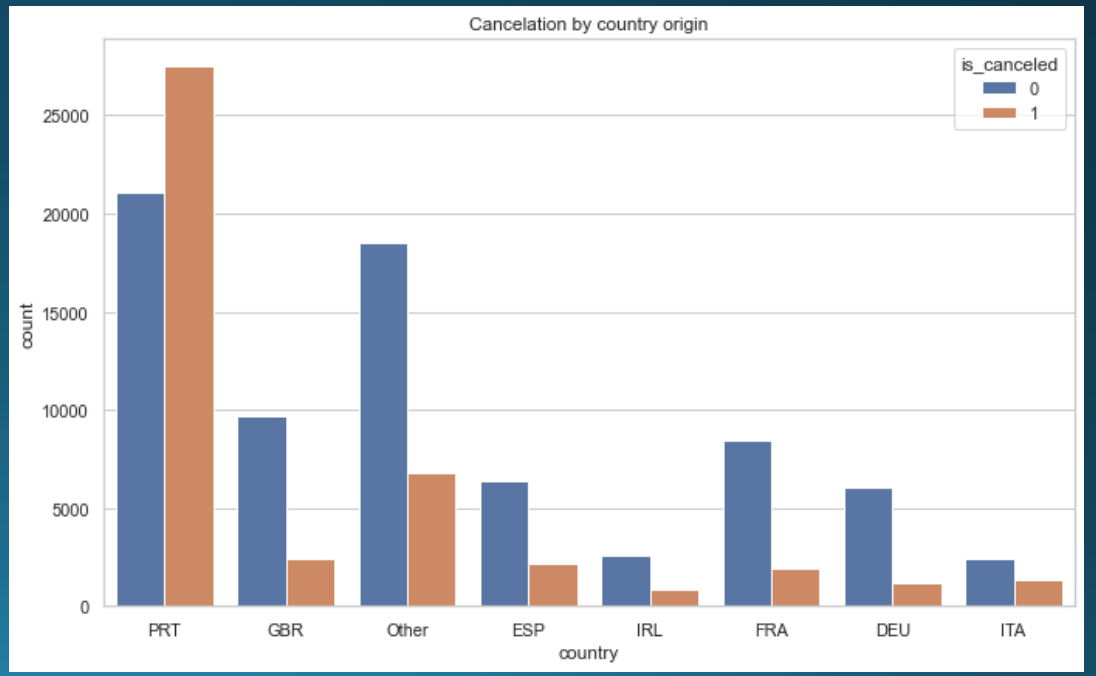
DATA LEAKAGE ANALYSIS

Data was collected with a “timestamp relative to the day prior to arrival date” - **Source Data Description**

“It is also common for hotels not to know the correct nationality of the customer until the moment of check-in” - **Source Data Description**

While more features were analyzed, we considered just the country origin to be source of data leakage and removed it.

Regarding possible Data Leakage from Train to Validation Sample, we made sure to make the train_test_split and fit Any feature engineering methods just on the training set.

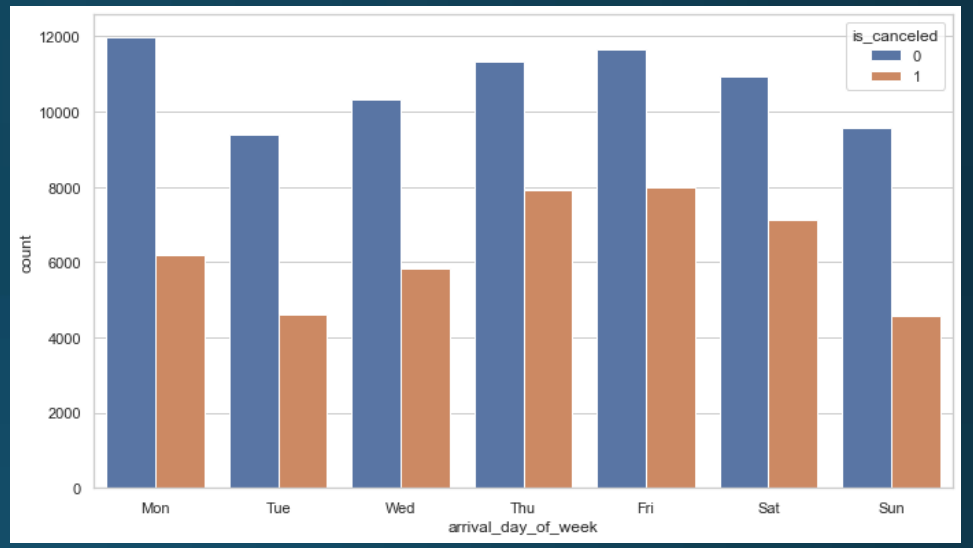


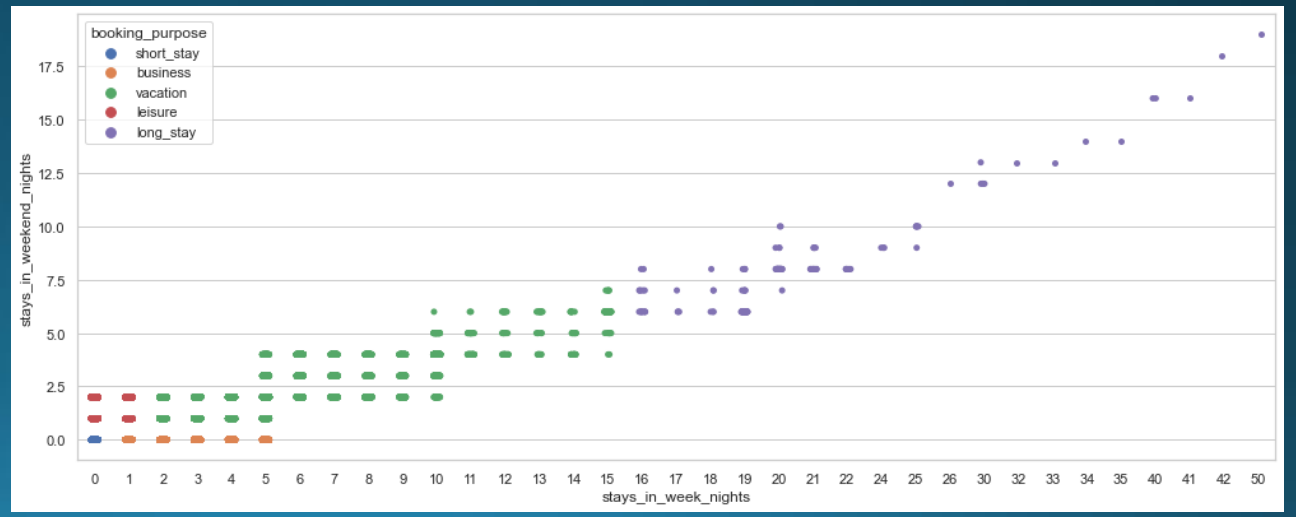
FEATURE ENGINEERING

Using the arrival date, we created a feature containing arrival's day of the week.

Using the number of weekdays and weekend days booked we clustered they stays in 5 categories based on duration, creating booking_purpose feature

Using the received room type and assigned room type we created the binomial feature received_different_room.



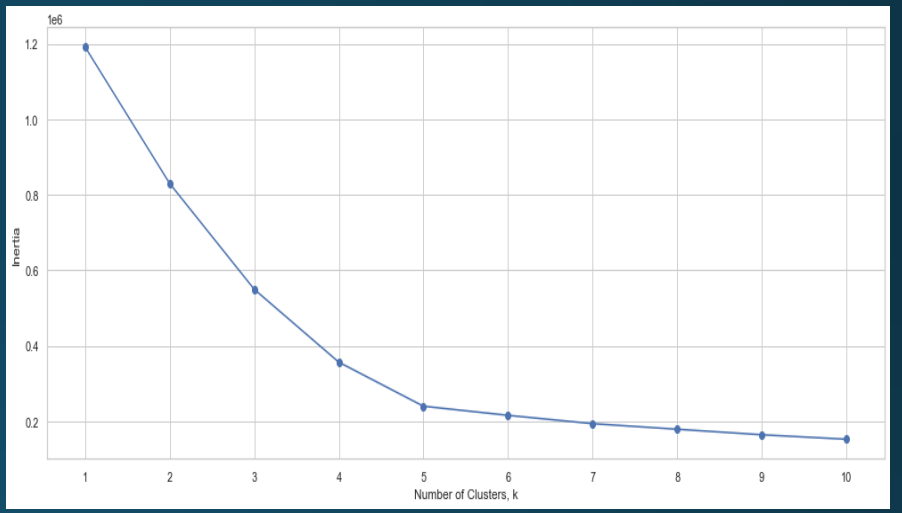


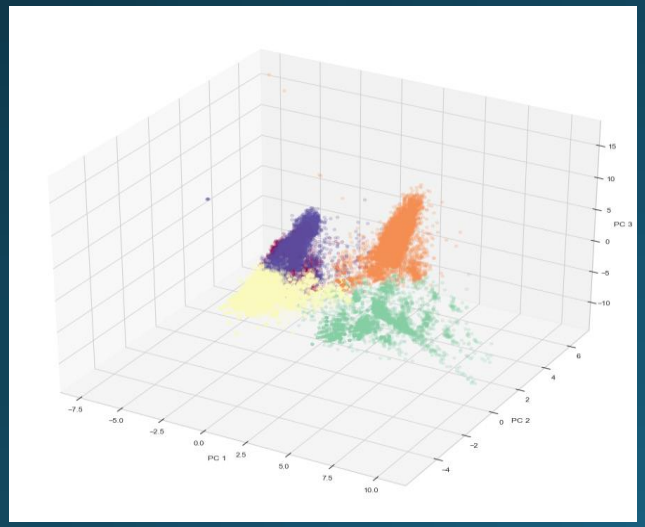
CUSTOMER SEGMENTATION – UNSUPERVISED LEARNING

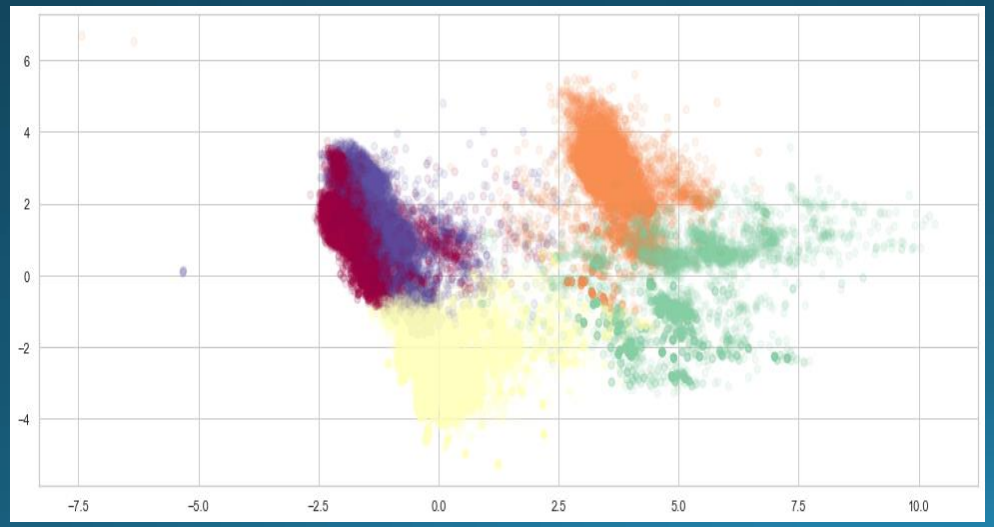
For the Unsupervised Learning Clustering we used just 21 features

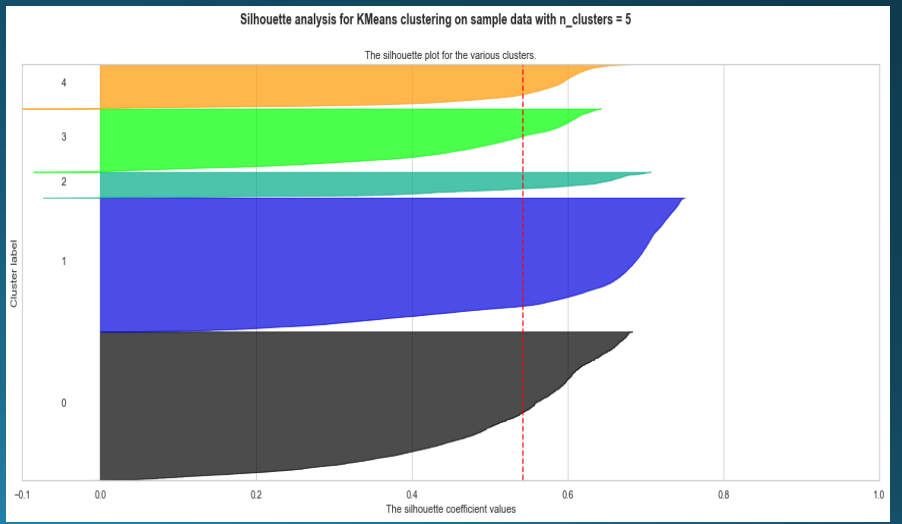
Compared how Kmeans model performed before and after scaling and PCA transformation

Best model with a silhouette_score of 0.54 returned 5 classes(with Scaler and 4PCA features)









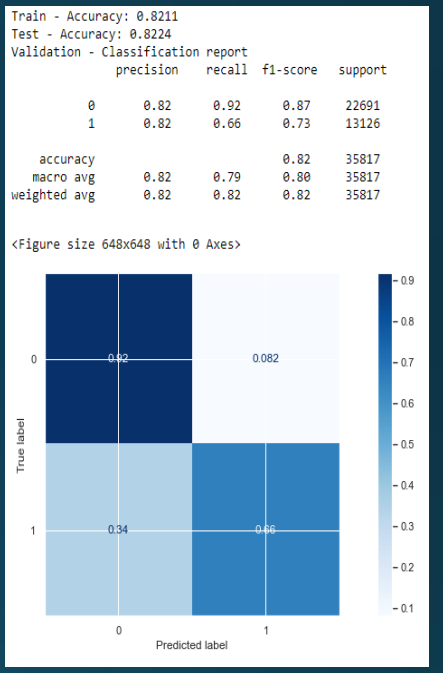
METRICS AND SUPERVISED LEARNING

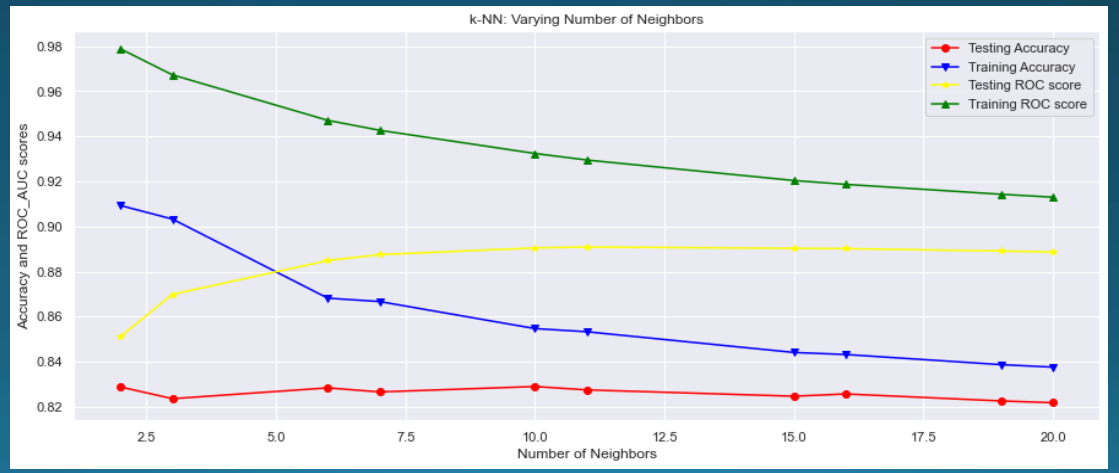
Metrics used: Accuracy and ROC's area under the curve for ranking and additional confusion matrix and classification report from visualization. We used a DataFrame to save the metrics.

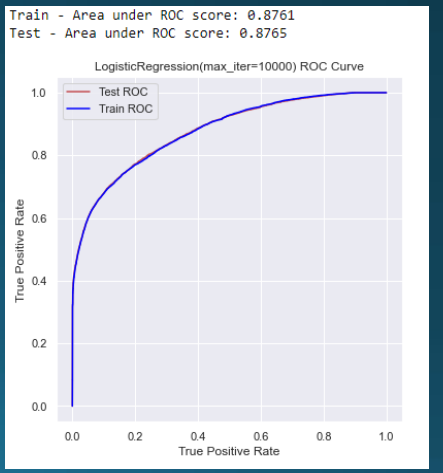
The models used are: LogisticRegression, KNN, Random Forest Tree and Catboost.

Other Tools used:

- Hyperparameter Tuning Using GridsearchCV and RandomSearchCV
- Regularization using Lasso or Ridge
- Feature Decomposition using PCA
- Feature Selection using permutation_importance and feature_importances_







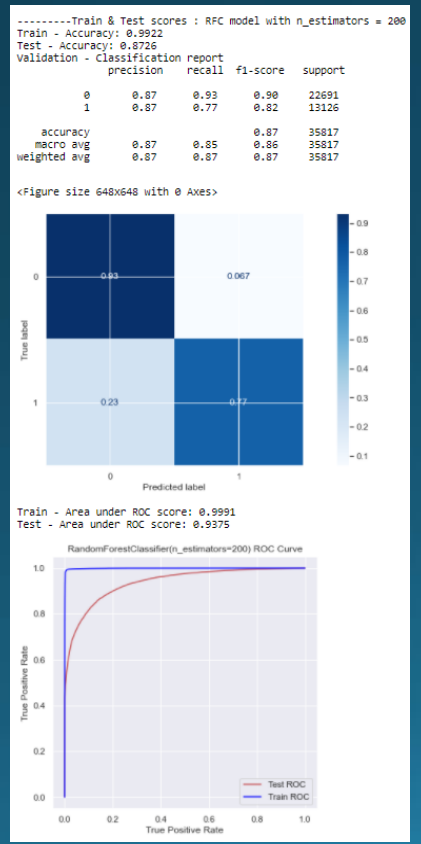
SCORES RANKING AND MODEL SELECTION

Best model – RFT with n_estimators = 200
Accuracy: 0.8726
roc_auc score: 0.9375

Second best model: Catboost
Accuracy: 0.8485

Third best: KNN(10 neighbors)
Accuracy: 0.8290
roc_auc score: 0.9324

Fourth best: LogReg(c=0.31)
Accuracy: 0.8225
roc_auc score: 0.8763



	model	train_accuracy	test_accuracy	train_ROC	test_ROC
28	(DecisionTreeClassifier(max_features='auto', random_state=1233902736), DecisionTreeClassifier(ma...	0.9922	0.8726	0.9991	0.9375
19	(DecisionTreeClassifier(max_features='auto', random_state=259966552), DecisionTreeClassifier(max...	0.9922	0.8724	0.9991	0.9371
27	(DecisionTreeClassifier(max_features='auto', random_state=64402675), DecisionTreeClassifier(max_...	0.9922	0.8718	0.9991	0.9369
26	(DecisionTreeClassifier(max_features='auto', random_state=1646429223), DecisionTreeClassifier(ma...	0.9920	0.8713	0.9991	0.9352
33	(DecisionTreeClassifier(max_features='auto', random_state=446869832), DecisionTreeClassifier(max...	0.9919	0.8694	0.9991	0.9332
34	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9256	0.8612	0.9853	0.9300
25	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9038	0.8604	0.9771	0.9323
24	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9034	0.8598	0.9764	0.9313
23	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9045	0.8592	0.9767	0.9300
31	(DecisionTreeClassifier(max_features='auto', random_state=446869832), DecisionTreeClassifier(max...	0.9909	0.8505	0.9988	0.9221
35	Catboost model	NaN	0.8485	NaN	NaN
32	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9199	0.8436	0.9840	0.9171
13	KNeighborsClassifier(n_neighbors=10)	0.8547	0.8290	0.9324	0.8905
9	KNeighborsClassifier(n_neighbors=2)	0.9093	0.8288	0.9788	0.8512
11	KNeighborsClassifier(n_neighbors=6)	0.8682	0.8284	0.9471	0.8849
29	(DecisionTreeClassifier(max_features='auto', random_state=446869832), DecisionTreeClassifier(max...	0.9867	0.8281	0.9981	0.8926
14	KNeighborsClassifier(n_neighbors=11)	0.8533	0.8275	0.9295	0.8909
12	KNeighborsClassifier(n_neighbors=7)	0.8667	0.8266	0.9427	0.8875
16	KNeighborsClassifier(n_neighbors=16)	0.8432	0.8257	0.9187	0.8902
30	(DecisionTreeClassifier(max_depth=20, max_features='auto',\n random_state=...	0.9327	0.8249	0.9884	0.8893
15	KNeighborsClassifier(n_neighbors=15)	0.8441	0.8247	0.9204	0.8903
22	(DecisionTreeClassifier(max_depth=10, max_features='auto', random_state=65203512), DecisionTreeC...	0.8257	0.8246	0.9092	0.9016
21	(DecisionTreeClassifier(max_depth=10, max_features='auto',\n random_state=...	0.8264	0.8245	0.9093	0.9015
20	(DecisionTreeClassifier(max_depth=10, max_features='auto',\n random_state=...	0.8249	0.8237	0.9064	0.8990
10	KNeighborsClassifier(n_neighbors=3)	0.9033	0.8236	0.9673	0.8699
17	KNeighborsClassifier(n_neighbors=19)	0.8387	0.8226	0.9143	0.8891
4	LogisticRegression(C=0.31622776601683794, max_iter=10000, solver='saga')	0.8208	0.8225	0.8760	0.8763
0	LogisticRegression(max_iter=10000)	0.8211	0.8224	0.8761	0.8765
6	LogisticRegression(C=0.01, max_iter=10000, penalty='l1', solver='saga')	0.8206	0.8223	0.8741	0.8745
1	GridSearchCV(cv=5, estimator=LogisticRegression(max_iter=10000),\n param_grid={'C': ...	0.8212	0.8222	0.8762	0.8766
7	LogisticRegression(C=0.31622776601683794, max_iter=10000, penalty='l1',\n solv...	0.8206	0.8222	0.8760	0.8764
8	LogisticRegression(C=10.0, max_iter=10000, penalty='l1', solver='saga')	0.8210	0.8221	0.8761	0.8765
5	LogisticRegression(C=10.0, max_iter=10000, solver='saga')	0.8210	0.8221	0.8761	0.8765
2	RandomizedSearchCV(estimator=LogisticRegression(max_iter=10000),\n param_distr...	0.8211	0.8221	0.8761	0.8765
18	KNeighborsClassifier(n_neighbors=20)	0.8376	0.8218	0.9130	0.8887
3	LogisticRegression(C=0.01, max_iter=10000, solver='saga')	0.8198	0.8216	0.8748	0.8752

RECOMMENDATIONS
MODEL IMPLEMENTATION AND USABILITY

One aspect had under consideration was to have the prediction as fast as possible.

Identified 2 possible models that can be deployed:

- *Front Desk Cancellations Percentage model.*
- *Monthly Cancellations Assessment.*

Manually created a new random booking and calculated the prediction.

The predicted probability is 0.52, so a 52% cancellation percentage.

```
new_booking
```

```
{'hotel': 'Algarve Resort Hotel',
 'arrival_date_year': 2019,
 'arrival_date_month': 7,
 'arrival_date_day_of_month': 12,
 'stays_in_weekend_nights': 4,
 'stays_in_week_nights': 5,
 'adults': 2,
 'children': 1,
 'babies': 0,
 'meal': 'BB',
 'market_segment': 'Direct',
 'distribution_channel': 'TA/TO',
 'is_repeated_guest': 1,
 'previous_cancellations': 1,
 'previous_bookings_not_canceled': 4,
 'reserved_room_type': 'A',
 'deposit_type': 'No Deposit',
 'agent': '250',
 'company': 'no_company',
 'arrival_day_of_week': 4,
 'arrival_date_week_number': 28,
 'booking_purpose': 'vacation'}
```

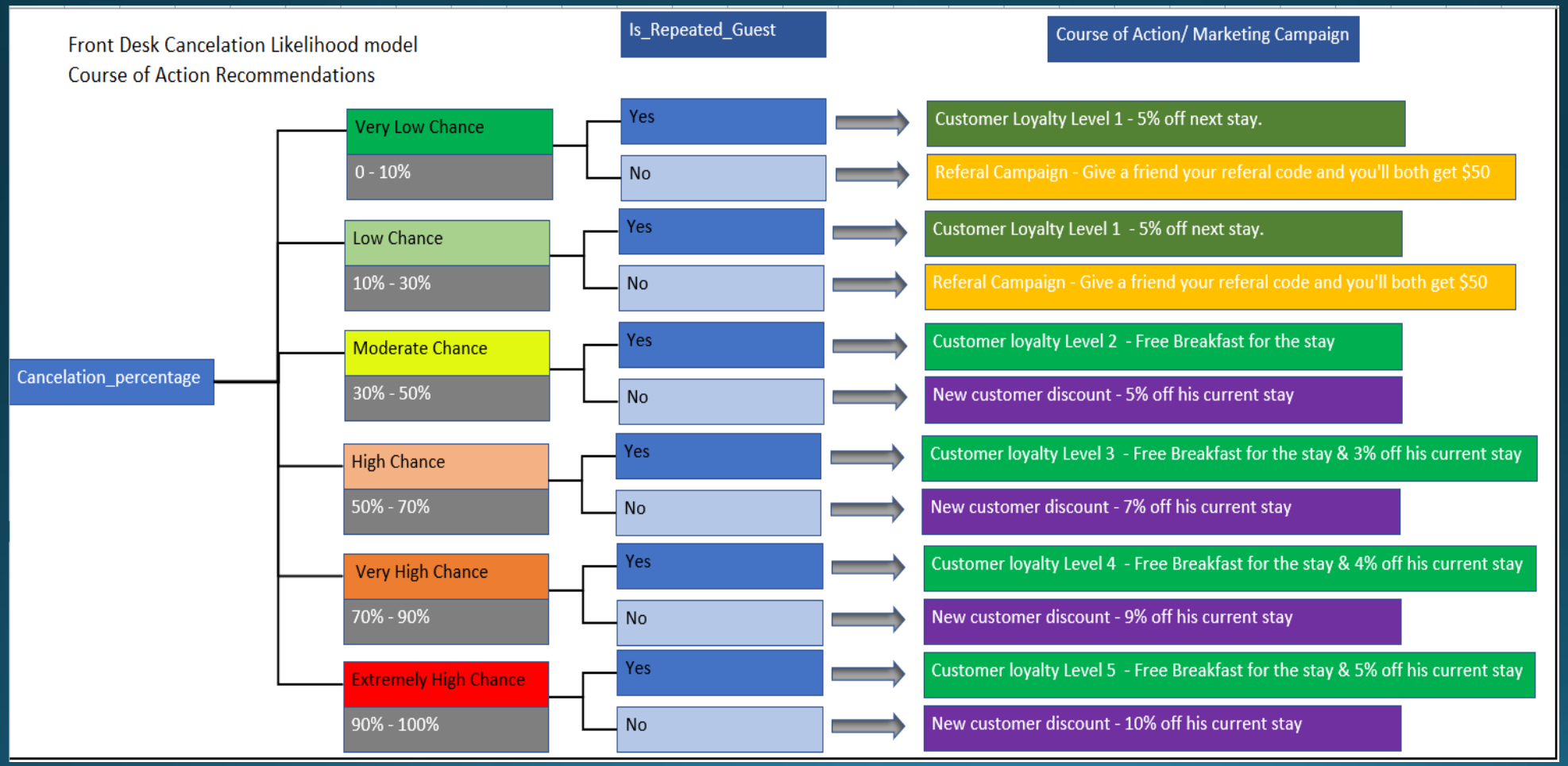
```
booking_prediction = best_model.predict(booking_w_dummy)
print('Customer is likely to cancel the booking' if booking_prediction[0]==1 else 'Customer is likely to keep the booking')
```

```
Customer is likely to cancel the booking
```

```
y_pred_prob = best_model.predict_proba(booking_w_dummy)[:,1]
print('Customer is {} % likely to cancel the booking'.format(round(y_pred_prob[0]*100,4)))
```

```
Customer is 52.0 % likely to cancel the booking
```

RECOMMENDATIONS
BUSINESS COURSE OF ACTION USING MODEL



FUTURE WORK

Are 99% of the customers with non-refundable full pay really canceling their booking?

Further Supervised Learning modeling using other Classifiers.

Create better user interactivity for adding a new booking.

After implementing the proposed Marketing Campaigns, we can gather data about the effectiveness of these Campaigns.

Then we can create other Machine Learning models that can predict what types of Marketing Campaigns we should use to target different groups of customers.

Thank you!

Questions?