האוניברסיטה העברית בירושלים

בית הספר למנהל עסקים

מדע המידע - 55866

תרגיל מסכם

מגישים:

Dana Benyamin Hassid 039759527
Katya Arnold 320608912
Egor Genov 208205831
Ran Amran 311332092
Binyamin Rosenstein 332599158

Final Project:

1. Research question:

Prediction of hotel booking cancellation probability, based on the booking details

We are proposing to build a business tool for the hotels that will allow them to predict the probability of a specific booking ending in order fulfillment or cancellation. It will provide hotel owners and management the opportunity to better allocate available budget and facilities based on these (aggregated) probability outcomes, and better prepare and manage both their waiting lists and overbookings.

2. Describing the dataset

The dataset is a single CSV file containing booking information about two hotels in Portugal: a city hotel and a resort hotel. The dataset is composed of 32 columns (16 integers, 12 objects and 4 floats) and 119390 rows (records). 79330 records correspond to the City hotel, and 40060 to the Resort hotel. It should be noted that the data is about only 2 hotels in one country, however the method and models we build should work on all hotels and in any country. Column explanations are presented in Index A. We are going to analyze the information and come up with a model that will be able to predict the probability of order cancellation.

3. Preparing the data

First we examine the data:

```
In [65]: hotels_df.head()
```

Out[65]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	sta
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	

3.1. Missing values:

We analyze the data for the missing values using the isna() function for this purpose:

```
In [43]:  #visualising missing data
          print("Data set missing data:\n", hotels_df.isna().sum())
```

```
Data set missing data:
 hotel                               0
is_canceled                          0
lead_time                            0
arrival_date_year                    0
arrival_date_month                   0
arrival_date_week_number             0
arrival_date_day_of_month            0
stays_in_weekend_nights              0
stays_in_week_nights                 0
adults                               0
children                             4
babies                               0
meal                                 0
country                            488
market_segment                       0
distribution_channel                 0
is_repeated_guest                    0
previous_cancellations               0
previous_bookings_not_canceled       0
reserved_room_type                   0
assigned_room_type                   0
booking_changes                      0
deposit_type                         0
agent                            16340
company                         112593
days_in_waiting_list                 0
customer_type                        0
adr                                  0
required_car_parking_spaces          0
total_of_special_requests            0
reservation_status                   0
reservation_status_date              0
dtype: int64
```

We imputed them with appropriate values (mode in case of categorical feature, and mean in case of numerical feature).

3.2. Total number of nights/guests is not null:

The total number of nights (combined week and weekend) reserved and the total number of guests included in the booking cannot be equal to zero. Therefore we removed entries where the number of weeknights and number of weekend nights add to zero, as well as those entries where the total number of guests are zero. The number of entries decreased to 118565.

```
In [49]:  #remove those rows where total number of nights is null
          hotels_df = hotels_df.loc[~nights_total]
          hotels_df
```

Out[49]:

s_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment	distribution_channel	is_repeated_guest	previous_cancc
0	1	1	0.0	0	BB	GBR	Direct	Direct	0	
0	1	1	0.0	0	BB	GBR	Corporate	Corporate	0	
0	2	2	0.0	0	BB	GBR	Online TA	TA/TO	0	
0	2	2	0.0	0	BB	GBR	Online TA	TA/TO	0	
0	2	2	0.0	0	BB	PRT	Direct	Direct	0	
...	...	...	...	...	...	...	...	...	...	
2	5	2	0.0	0	BB	BEL	Offline TA/TO	TA/TO	0	
2	5	3	0.0	0	BB	FRA	Online TA	TA/TO	0	
2	5	2	0.0	0	BB	DEU	Online TA	TA/TO	0	
2	5	2	0.0	0	BB	GBR	Online TA	TA/TO	0	
2	7	2	0.0	0	HB	DEU	Online TA	TA/TO	0	

```
In [170]:  # check if there are rows where the total number of guests is null
           guests_total = (hotels_df.children == 0) & (hotels_df.adults == 0) & (hotels_df.babies == 0)
           hotels_df[guests_total]
```

Out[170]:

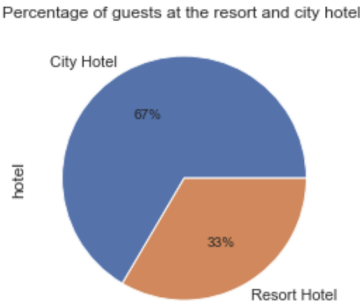
ys_in_week_nights	adults	children	babies	meal	country	market_segment	distribution_channel	is_repeated_guest	previous_cancellations	previous_bookings_not
3	0	0.0	0	SC	PRT	Corporate	Corporate	0	0	
2	0	0.0	0	SC	ESP	Groups	TA/TO	0	0	
4	0	0.0	0	SC	PRT	Groups	TA/TO	0	0	
4	0	0.0	0	SC	PRT	Groups	TA/TO	0	0	
8	0	0.0	0	BB	PRT	Direct	Direct	0	0	
...	...	...	...	...	...	...	...	...	...	...
3	0	0.0	0	BB	CHE	Online TA	TA/TO	0	0	
1	0	0.0	0	SC	PRT	Complementary	Direct	0	0	
1	0	0.0	0	SC	SWE	Online TA	TA/TO	0	0	
5	0	0.0	0	SC	RUS	Online TA	TA/TO	0	0	
2	0	0.0	0	BB	BRA	Offline TA/TO	TA/TO	0	0	

4. EDA

4.1. Visualizing of the columns data:

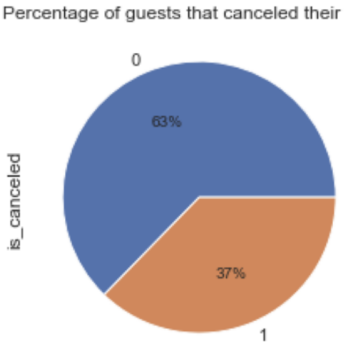
In order to analyze the data we create graph visualization of part of the columns:

4.1.1. Percentage distribution of observations by hotel:



as we can see from the graph above 67% of the observation is of the city hotel, that's why we need to create a heatmap on both hotel's so we can check if this information affect the target variable

4.1.2. Percentage of guests that canceled their order

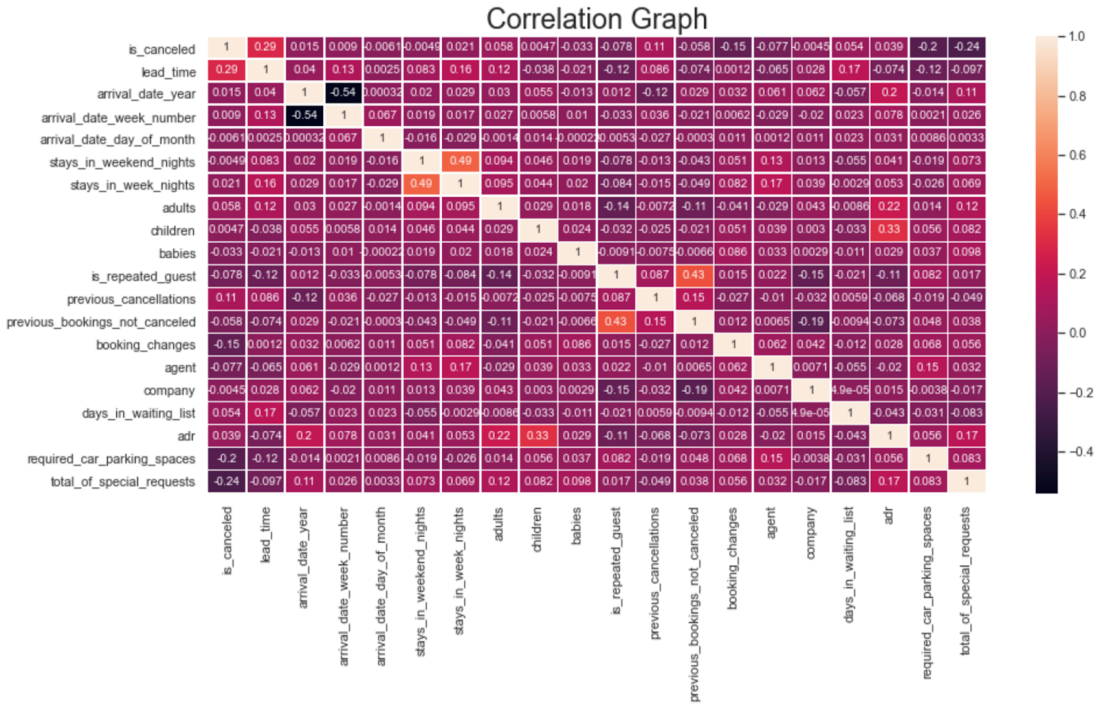


We also examined the amount of reservations that has been canceled for each one of the hotel's:

4.2. Heat map:

As noted above, we made a heat map for each hotel to see which of the attributes are highly correlated with the "is canceled" attribute and if there is a difference between the hotels in terms of important features.

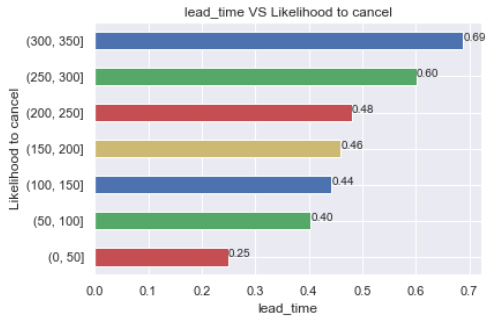
We can see that the highest correlated attributes to 'is canceled' are the same for both hotels and the attributes are: "lead_time", "previous_cancelations", "booking_changes" "required_car_parking_spaces", "total_of_special_requests".



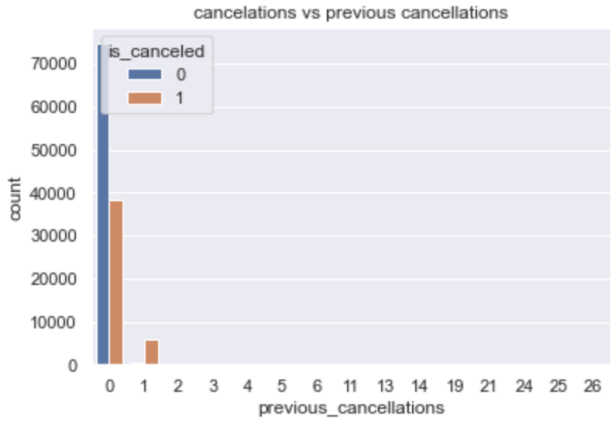
Therefore we choose not to separate the analysis between the hotels; we will attempt to better understand the correlations of the most relevant attributes by performing EDA on the full dataset:

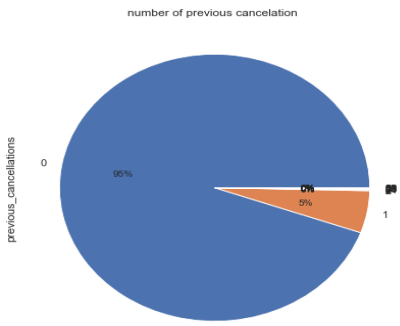
4.2.1. Lead time:

We can see from the plot that the chance of cancellation is higher the longer in advance the reservation was booked.



4.2.2. Previous cancellation:

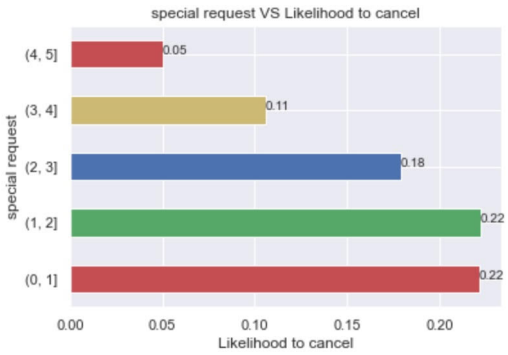




We can see that although we have a low number of observations where the booking was cancelled by a customer who had done so before, the correlation is relatively high--meaning that a customer who has previously cancelled is very likely to cancel again--and therefore we decided not to drop it.

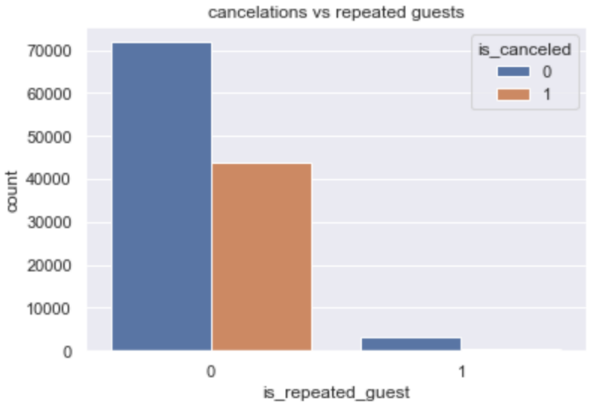
4.2.3. Total number of special requests:

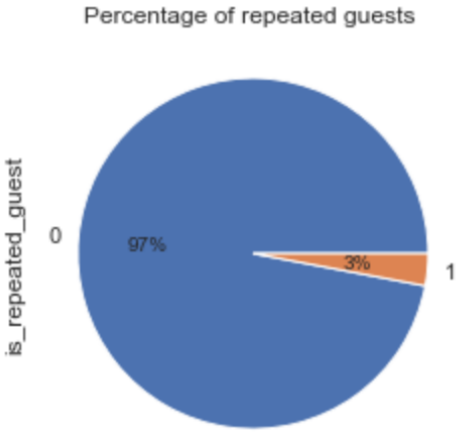
We can see that as the number of special requests rises the chances of cancelation decrease.



4.2.4. is repeated guest:

Additionally, it seems prudent to check if a repeat guest--one who has stayed at the hotel before--is more or less likely to cancel.





4.3. While doing EDA we noticed more preparations needed to make the data more accessible and useable:

- the "reservation_status_date" column.
  For convenience we decided to break this column into 3 - year, month and day. Then we ran a correlation between the 3 values and the "is_cancelled" column and saw that the correlation of day is very low that's why we dropped it. This manipulation caused us to add 3 columns and drop 1. So in the end we remained with 34 columns.

- drop all the irrelevant columns:
  - Due to low correlation we dropped the following columns: 'days_in_waiting_list', 'arrival_date_year', 'assigned_room_type', 'booking_changes', 'country', 'days_in_waiting_list', 'res_status_day'.

- We can also drop the 'reservation_status' column because we have the `is_cancelled` column that we can use instead.

So, now, we have 26 columns (one of them is decision vari).

4.4. Dummy variables.

We have 9 columns with categorical values that must be converted to integer values.
The columns are: 'hotel', 'meal', 'market_segment', 'distribution channel',
'reserved room type', 'deposit type', 'customer type', 'arrival date month', 'reservation status'.
We preferred the dictionary method over the pandas built-in pandas.get_dummies( ) because of lower output complexity and because our problem is a classification problem with a categorical (represented by a dummy variable) target variable, and therefore we do not expect linear correlations between the dummy value and the independent variables.
To do so we used this code:

Dummy variables

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

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

hotels_df['market_segment'] = hotels_df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/
                                                               'Complementary': 4, 'Groups': 5, 'Undefined': 6, 'Aviation':

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

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

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

hotels_df['customer_type'] = hotels_df['customer_type'].map({'Transient': 0, 'Contract': 1, 'Transient-Party': 2, 'Group
hotels_df['arrival_date_month'] = hotels_df['arrival_date_month'].map({'January': 1, 'February': 2, 'March': 3, 'April':
```

4.5. normalizing variables to the same scale:

We wanted the values to be in the same range, so we ran the variance function and decide to run a log function to make the same range.

```
#normalizing variables to the same scale
hotels_df['lead_time'] = np.log(hotels_df['lead_time'] + 1)
hotels_df['arrival_date_month'] = np.log(hotels_df['arrival_date_month'] + 1)
hotels_df['arrival_date_week_number'] = np.log(hotels_df['arrival_date_week_number'] + 1)
hotels_df['arrival_date_day_of_month'] = np.log(hotels_df['arrival_date_day_of_month'] + 1)
hotels_df['agent'] = np.log(hotels_df['agent'] + 1)
hotels_df['company'] = np.log(hotels_df['company'] + 1)
hotels_df['adr'] = np.log(hotels_df['adr'] + 1)
hotels_df['res_status_month'] = np.log(hotels_df['res_status_month'] + 1)
```

Before running the models, we check one last time if there were any nan / null values.
We discovered there was just one such row - and removed it.
Now we can start the models:

5. Model building:

In order to predict the outcome of the booking we decided to employ a few supervised learning ML models and then compare between them for the best results as well as reduce the possibility of overfitting. Since our problem is a classification problem, we will stick to classification models.

To evaluate the models we used the following measures:

- *Accuracy = correct predictions / total number of data points* - Fraction of predictions our model got right
- $Precision = TP/(TP+FP)$ - Precision is the fraction of correctly predicted classes
- $Recall = TP/(TP+FN)$ - is the fraction of classes which are relevant to the query and were successfully retrieved.
- *F1 score* - a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.
  F1 = 2 * (precision * recall) / (precision + recall)
- ROC/AUC score - demonstrates to what extent the model is capable of distinguishing between classes at given threshold settings. ROC is a probability curve (a graphical representation of (1-specificity) on the X axis and the true positive rate (sensitivity or 1 - the false negative rate) on the Y axis) and AUC represents the degree or measure of separability.

5.1. Logistic regression model

Logistic regression is a statistical model that allows us to estimate the probability of certain events. It is based on the logistic function to model a binary dependent variable. In this project we model the probability of a dependent variable ("is_cancelled") to equal 1.

the results are:

```
Accuracy Score of Logistic Regression is : 0.7945459657014338
Confusion Matrix :
[[20907  1483]
 [ 5825  7355]]
Classification Report :
              precision    recall  f1-score   support

           0       0.78      0.93      0.85     22390
           1       0.83      0.56      0.67     13180

    accuracy                           0.79     35570
   macro avg       0.81      0.75      0.76     35570
weighted avg       0.80      0.79      0.78     35570
```

5.2. K-nearest neighbours

The k-nearest neighbors algorithm is a non-parametric classification method that uses k closest points to perform classification. As the output the model is classifying the test set to one of the known classes of the training set. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that

single nearest neighbor. In our model we used *sklearn.neighbors.KNeighborsClassifier()* function with its default settings with n_neighbors=5.

```
Accuracy Score of KNN is : 0.845178521225752
Confusion Matrix :
[[20361  2029]
 [ 3478  9702]]
Classification Report :
              precision    recall  f1-score   support

           0       0.85      0.91      0.88     22390
           1       0.83      0.74      0.78     13180

    accuracy                           0.85     35570
   macro avg       0.84      0.82      0.83     35570
weighted avg       0.84      0.85      0.84     35570
```

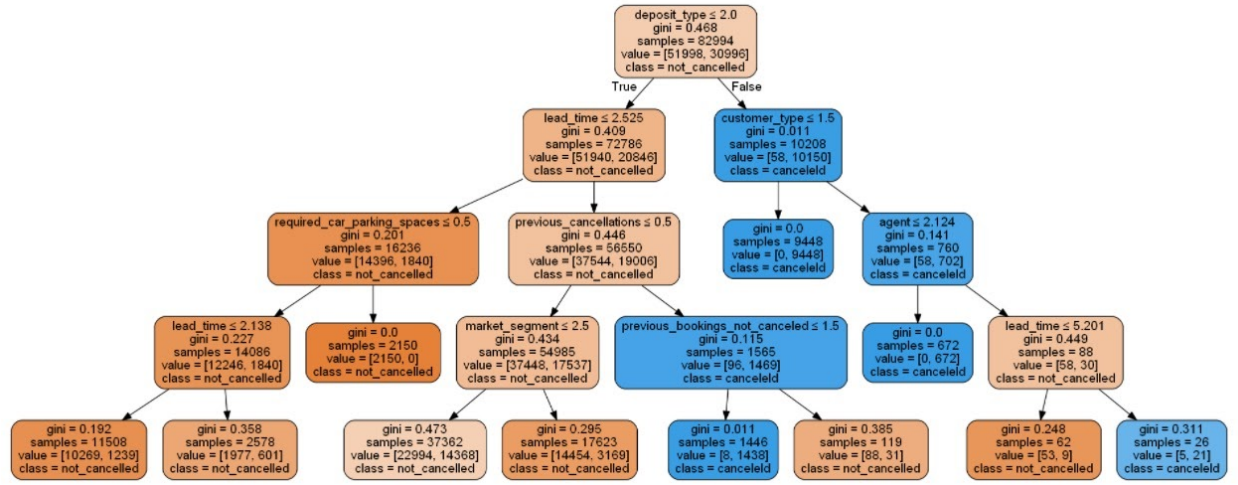
5.3. Decision tree classifier

Decision Tree is a predictive modelling approach that uses a decision tree to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). As we are not sure that the correlations found from the data are linear and not all variables are boolean, we chose to employ the tree classification model, which is meant to deal with this kind of problem. If compared to logistic regression, the tree classifier can be used for higher-dimensional data, since it bisects the space into smaller and smaller regions, whereas logistic regression fits a single line to divide the space exactly into two. The dependent variable is categorical, therefore the classification tree was chosen over the regression tree. We used *sklearn.tree.DecisionTreeClassifier()* function with its default parameters (gini impurity criterion, best split, max_depth = 4).

```
Accuracy Score of Decision Tree is : 0.9121169524880517
Confusion Matrix :
[[20797  1593]
 [ 1533 11647]]
Classification Report :
              precision    recall  f1-score   support

           0       0.93      0.93      0.93     22390
           1       0.88      0.88      0.88     13180

    accuracy                           0.91     35570
   macro avg       0.91      0.91      0.91     35570
weighted avg       0.91      0.91      0.91     35570
```



5.4. Random forest classifier

Random forest classifier is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. If compared to the classification tree, it reduces variance of prediction by combining many independent classifiers and thereby prevents data overfitting as well as identifying the most significant variables.

In the current project we used *sklearn.tree.DecisionTreeClassifier()* function with its default parameters. In the current project we used *sklearn.ensemble.RandomForestClassifier()* function with its default parameters.

```
Accuracy Score of Random Forest is : 0.9385156030362665
Confusion Matrix :
[[21989   401]
 [ 1786 11394]]
Classification Report :
              precision    recall  f1-score   support

           0       0.92      0.98      0.95     22390
           1       0.97      0.86      0.91     13180

    accuracy                           0.94     35570
   macro avg       0.95      0.92      0.93     35570
weighted avg       0.94      0.94      0.94     35570
```

6. Model comparison

In order to compare between the models we looked at the accuracy score and ROC-AUC score. Those scores are the representative characteristics for evaluation of classification models. Accuracy score represents the fraction of correct predictions and is calculated as number of correct predictions /total number of predictions.

ROC-AUC score demonstrates how much the model is capable of distinguishing between classes.

	Model	Score
3	Random Forest Classifier	0.938516
2	Decision Tree Classifier	0.912117
1	KNN	0.845179
0	Logistic Regression	0.794546

7. Conclusion

- Our goal was to build a model that will be able to predict the probability of order cancellation based on the booking details. We built four different models, employing logit, decision tree classifier, random forest and KNN algorithms.
- We conclude that the Random forest classification model performs the best for this type of task. It has highest accuracy score (the fraction of correct predictions is the highest) and ROC/AUC score (capability of model to distinguish between the classes)
- We would require more data from more hotels and more countries in order to make more reliable predictions that could be sold like a booking evaluation tool.
- By performing EDA we found that there is a high chance of cancellation when the customer had previous cancellations; additionally, being a repeated guest is a factor promoting a higher probability of ordering. As the dataset contained relatively few observations containing these features, additional data on these cases--repeated guests and cancellation history--would enhance the accuracy of the model.
- Directions for future analysis: we propose to continue the research by adding more features to the data. For example, we could require data on hotel quality, availability of different meal preferences, stay prices, pool availability, hotel type, resort availability, attractions etc. This data can add significantly to the model precision and therefore increase the marketing value of the product.

Despite the fact that the model could be improved through enhancement of the dataset--by adding hotels, observations, features, etc.--our work still provides a predictive model with an accuracy of 93.8% accuracy--much better than the “flip of a coin.”

Index A:

column	name	explanation
1	hotel	two unique values: City Hotel, Resort Hotel.
2	is_canceled	Value indicating if the booking was canceled (1) or not (0)
3	lead_time	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
4	arrival_date_year	Year of arrival date
5	arrival_date_month	Month of arrival date
6	arrival_date_week_number	Week number of year for arrival date
7	arrival_date_day_of_month	Day of arrival date
8	stays_in_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
9	stays_in_week_nights	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
10	adults	Number of adults
11	children	Number of children
12	babies	Number of babies
13	meal	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
14	country	Country of origin. Categories are represented in the ISO 3155–3:2013 format
15	market_segment	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
16	distribution_channel	Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”
17	is_repeated_guest	Value indicating if the booking name was from a repeated guest (1) or not (0)
18	previous_cancellations	Number of previous bookings that were cancelled by the customer prior to the current booking

19	previous_bookings_not_canceled	Number of previous bookings not cancelled by the customer prior to the current booking
20	reserved_room_type	Code of room type reserved. Code is presented instead of designation for anonymity reasons.
21	assigned_room_type	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons.
22	booking_changes	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
23	deposit_type	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories: No Deposit – no deposit was made; Non Refund – a deposit was made in the value of the total stay cost; Refundable – a deposit was made with a value under the total cost of stay.
24	agent	ID of the travel agency that made the booking
25	company	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
26	days_in_waiting_list	Number of days the booking was in the waiting list before it was confirmed to the customer
27	customer_type	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it; Group – when the booking is associated to a group; Transient – when the booking is not part of a group or contract, and is not associated to other transient booking; Transient-party – when the booking is transient, but is associated to at least other transient booking
28	adr	Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
29	required_car_parking_spaces	Number of car parking spaces required by the customer
30	total_of_special_requests	Number of special requests made by the customer (e.g. twin bed or high floor)
31	reservation_status	Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why
32	reservation_status_date	Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel