Introduction

The real power of Machine Learning comes from the fact that it is a technology that can enable the prediction of future events even when certain datasets are absent. There are many ways to help and improve business models in a wide range of industries. In this project we will concentrate on the hospitality industry. Some machine learning models are already used, such as "The LightStay" program at Hilton which predicts energy, water, and waste usage and costs. Another example is the online travel agency sites like Expedia that uses Machine Learning to learn the preferences of customers' travel and then recommend hotels and flights according to these preferences.

The data used in our project is data of hotel reservations. Our main target will be to predict whether a reservation will be cancelled or not, whenever given a new hotel reservation.

Part 1 - data Exploration

After implementing technical functions such as importing libraries we will work with, loading the data and merging it with the labels, we will start exploring our data as it is.
We will try to understand what every feature describes, what is the type of the feature and how it is distributed by simple functions such as describe(), dtypes, head() and keys(). As described in table 1.1, The data has 33 features from various types: categorical, numerical and boolean. Also, there are 13 Anonymous features which we will have to explore and understand better. The anonymous features can be anything, from arrival date to type of meal. There are many things we can do to get some clues about these features. We can check their histograms or density, types of values and more. By looking at the correlation plot as described in figure 1.1 we can see correlations between time_until_order and anon_feat_11, and anon_feat_13 and cancelation. Another useful plots to understand the data better are cancelation per feature plot (1.2), missing values bar plot (1.2) and feature density plot

Part 2 - Preprocessing & Preparation

Starting with one of the most common problems when dealing with data - missing values. Missing values pose an important challenge, because typical modeling procedures simply discard these cases from the analysis. As we can see on figure 1.2, company (94.35%) and anon_feat_13 (93.55%) have the largest count of NaN values. Natural choice will be dropping these features, another option will be to fill the NaN's values. The second option can cause noise or even bias toward the filled value, and that will cause more damage than some good impact on our data. Due to the large percentage of the missing values, we find it more appropriate to use the first method - dropping company and anon_13 features.

Also, we can see in (zero percent) we have many features with more than 90% zero values of the data. We have decided these features are not useful enough because of the low variance the zeros cause. As a result, 'babies' and 'children' will be replaced by a new feature, as well as 'prev_canceled' and 'prev_not_canceled'. Later on we will explain about a new feature 'family' which we will decide to use instead of 'children'. Moreover, 'anon_feat_3' and 'anon_feat_6' will be deselected on the feature selection part. - why?

Regarding the other features, we used the second method of filling the missing values. For numerical features we have replaced the missing values with the mean value of the feature while in categorical features we replaced it with the word 'Other'. This method is based on the assumption that the percentage of the missing values is relatively negligible and will cause very small noise.

Next, we will try to detect outliers. In statistics, an Outlier is an observation point that is distant from other observations. There are different types of outliers in our data. The first type of outlier complies with the definition above and we can detect this type of outlier as the one that appears in the boxplot as points (or circles) . By plotting many boxplots for numerical data, we can see on figure 1.3 that there are indeed some outliers to remove. We can see that 'adr', 'changes' and 'children' are the main features with a big amount of outliers. The challenge we are facing is that we do not want to remove these observations due to the big amount (adr - 15,666). By trying to replace the 'adr' outliers with the 'adr' causes a lot of noise, So we after some attempts applying the model with and without the outliers we realised that we are getting better results when leaving the outliers as they are. We have decided to not include 'changes' when we will implement the feature selection.

The second type of outliers will be singular observations. For example, if we have a minor number of reservations from Israel and it has been canceled, it can affect the prediction because we do not have more data about hotel reservations from 'order type undefined'. Due to the fact we want to avoid removing entire observations, we have saved these categories and after transforming them into dummy variables, we dropped some of them because of large amount of zeros.

Furthermore, we have created many new features. Most of the new features are derived from the data, and there is one feature which was extracted from an external source.

1. Season - we created this feature assuming that there are no northern and southern sides of the earth and the seasons are in constant months. Thus, between x-y the season is summer and between x-y it is winter.
2. Family - is defined as the total number of adults, babies and children. Assuming that if there are children and babies in the reservation with adults they are considered as one family.
3. GDP - by loading an external dataset of countries' economic data, and merging by the name of the country, we added a new feature of the GDP of the country. GDP is Gross domestic product. It is the total monetary or market value of all the finished goods and services produced within a country's borders in a specific time period.
4. Dummy features - we handle categorical features with dummy variables. Every such feature has the name of the category it represents and if a reservation describes the feature it will be signed as 1, otherwise 0.
5. New customer - After identifying that 'prev_canceled' and 'prev_not_canceled' have above 95% zero values, we decided to create this new feature to spot when an observation is for a new customer or not. If both 'prev_canceled' and 'prev_not_canceled' are equal to zero, then it means the specific reservation belongs to a new customer.

Finally, after the data is tidy and contains only numerical data, we will apply the MinMaxScaler function in order to normalize our data. The goal of normalization is to change the values of numeric columns in the dataset to a common scale (in our case between 0 and 1), without distorting differences in the ranges of values. Normalization is useful when the data has varying scales and the algorithm used does not make assumptions about the distribution of the data, such as k-nearest neighbors and neural networks which we will implement soon.

After cleaning and normalizing the data there is another important thing to be considered - the dimensionality of the data. Having a large number of dimensions can dramatically impact the performance of machine learning algorithms fit on data with many input features. After a certain point, increasing the dimensionality of the problem by adding new features would actually degrade the performance of the classifier. Therefore, it is often desirable to reduce the number of input features, by manual feature selection or by applying PCA algorithm. We chose manual feature selection and decided to drop X features, for different reasons: big percentage of missing values or zeros, to avoid increasing dimensionality because of creating many new dummy features for every different category. Full explanation about every dropped feature can be seen on table 2.x.

Part 3 - Modeling

Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions. There is a big variety of different machine learning models.We chose to examine in our project four of them, two sample models and two advanced models. For choosing the parameters for every model we used GridSearchCV. The GridSearch is fitting on a dataset all the possible combinations of parameter values which are declared by us, and the best combination is retained.

The first model is **KNN- k nearest neighbors**. This is an easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The best choice of k depends upon the data. generally, larger values of k reduces the noise, but can make boundaries between classes less distinct. Our choosing of k and other parameters based on hyperparameter optimization by GridSearchCV as described in table 3.x. we got a result of 86.157% AUC.

The second model is **Logistic regression**. In this model Input values are combined linearly using weights or coefficient values to predict an output value. we got a result of 85.112% AUC.

Moving up to the advanced models, the third model used is **Random Forest Classifier**. We got a result of 92.341% AUC.
Every parameter affects differently : Larger trees can fit more complex functions, but also increase the ability to overfit. Tree size can improve performance by balancing between over- and underfitting. Big number of trees in the forest decreases the variance of the overall model, and doesn't contribute to overfitting.

The last model used is **MLP - Multilayer perceptron**.It is a supplement of feed forward neural network. It consists of three types of layers. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.It can distinguish data that is not linearly separable. We got a result of 91.336% AUC.

Chapter 4 - Metrics

Without doing a proper evaluation of the models using different metrics, and depending only on accuracy, it can lead to a problem when the respective model is deployed on unseen data and can result in poor predictions. In this section we will cover some of the most useful metrics.

Firstly, we will plot the confusion matrix as it is described on figure 4.1. Confusion Matrix table is often used to describe the performance of a classification model. It contains 4 types of measurements: TP,TN, FP, FN.For example, In our case - the random forest model, as we can see on Figure 4.1, 9,661 predictions are signed as TP - we predicted the observation is 1, and we were right. 4,693 predictions are signed as TN - we predicted the observation is 0, and we were right. 837 predictions are signed as FP - we predicted the observation is 1, and we were wrong. In our case 1,653 predictions are signed as FN - we predicted the observation is 0, and we were wrong. Our goal is to have the highest values of TP-TN diagonal line, and the lowest values in the other lines – FP-FN.

Next measurement will be Receiver Operating Characteristics (ROC) curve.This curve plots TPR vs. FPR values (which are based on calculation of TP,TN, FP, FN mentioned above) at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. Figure 4.x shows the ROC curves for every model we have tested.The Area Under That Curve is also important measurement and the Higher (maximum 1) the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.

Moving up to Cross-validation, It shuffles the given data randomly and for every fold it splits the data differently into train and test. We have decided to choose k=10 within a consideration of time complexity and accurate results. The function plots for each iteration the ROC curve and an additional mean curve after all the iterations.
We have ran the K-fold function on all of our models and received the following results-

- Knn - 84.4%
- Logistic Regression - 83.6%
- MLP - mean auc 89.7%
- Random Forest -mean auc 91.5%

Conclusions

Appendix

Chapter 1 - data Exploration

Table 1.1 - Description of Dataset

Feature name	type	values	description
time_until_order	float		
order_year	int	2015-2017	
order_month	object	January-December	
order_week	object	week_1 - week_52	
order_day_of_month	int	1-31	

adults	int	1-50	Number of adults
children	float	0.0, 1.0, 2.0, 3.0, 10.0	Number of children
babies	int	0,1,2,10	Number of babies
country	object	Three letter encoded country (‘ISR’)	Country of origin.
order_type	object	'Offline TA/TO' 'Online TA' 'Direct' 'Corporate' 'Groups' 'Aviation' 'Complementary' 'Undefined'	“TA” means “Travel Agents” and “TO” means “Tour Operators”
acquisition_channel	object	'TA/TO' 'Direct' 'Corporate' 'GDS' 'Undefined'	
prev_canceled	int	0-21	Number of previous bookings that were cancelled by the customer prior to the current booking
prev_not_canceled	int	0-71	Number of previous bookings not cancelled by the customer prior to the current booking
changes	float	0.0-21.0	Number of changes/amendments made to the booking from the moment the booking
deposit_type	object	'No Deposit' nan 'Non Refund' 'Refundable'	Indication on if the customer made a deposit to guarantee the booking

agent	float		ID of the travel agency that made the booking
company	float		ID of the company/entity that made the booking or responsible for paying the booking.
customer_type	object	'Transient-Party" Transient' 'Contract' nan 'Group'	
adr	float		Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
anon_feat _0,5,6,7,9,10,11,13	float		
anon_feat _1-4, 8	int		
anon_feat_12	bool	True False	

Figure 1.1 - Correlation plot between features

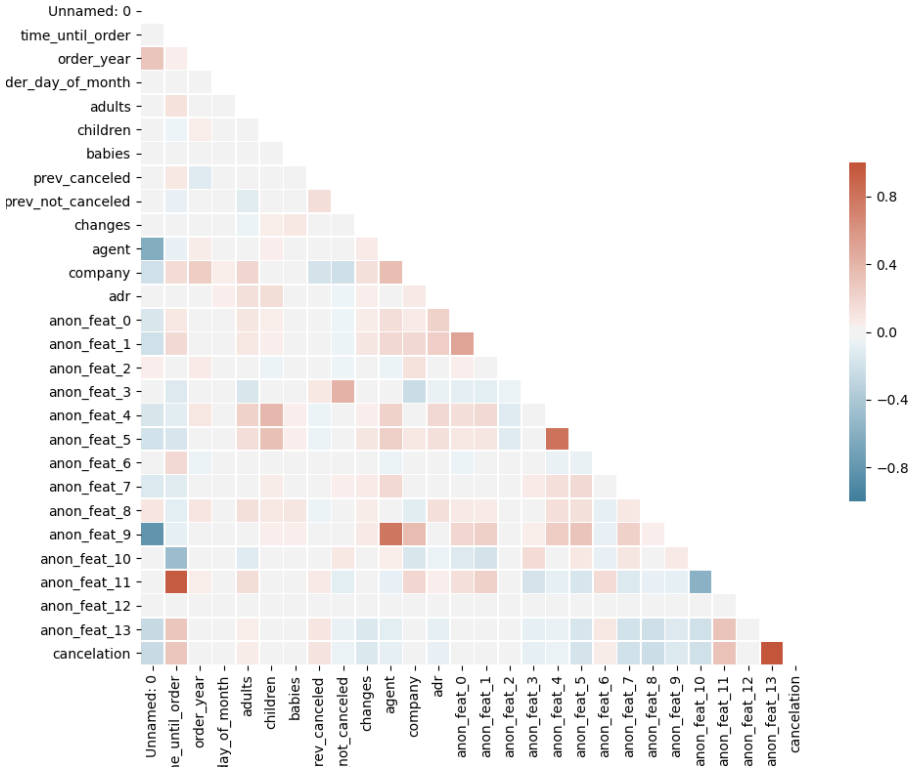


Figure 1.2 - Cancelation per order type

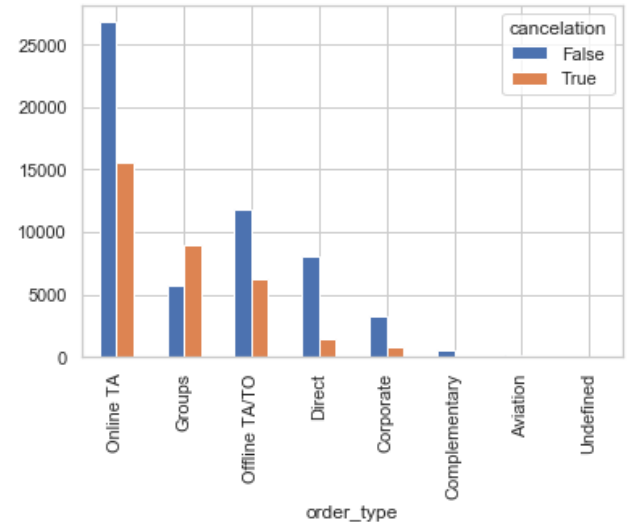
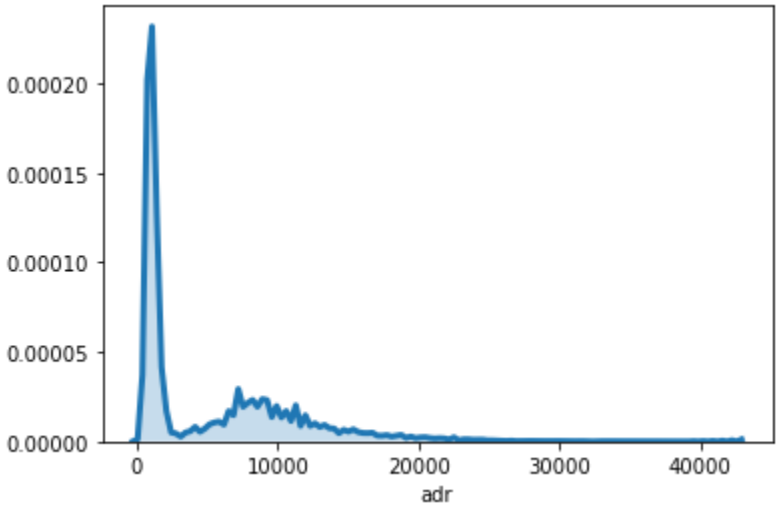


Figure 1.3 - Feature density plot



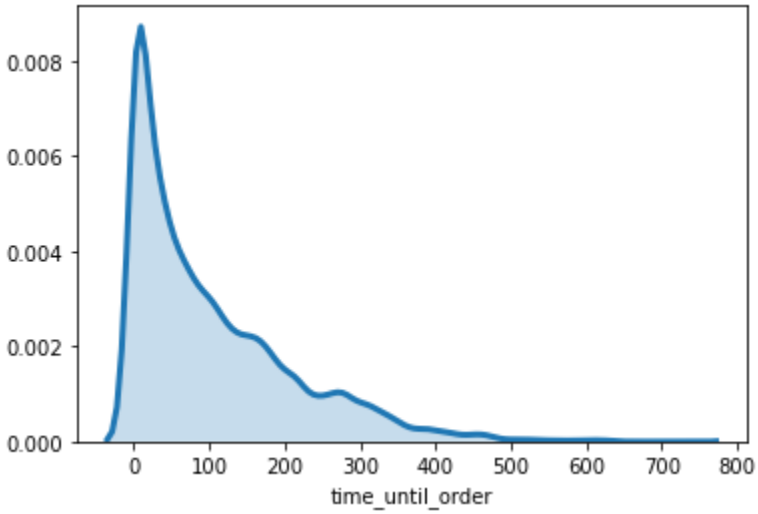


Figure 1.4 - Nan Counts

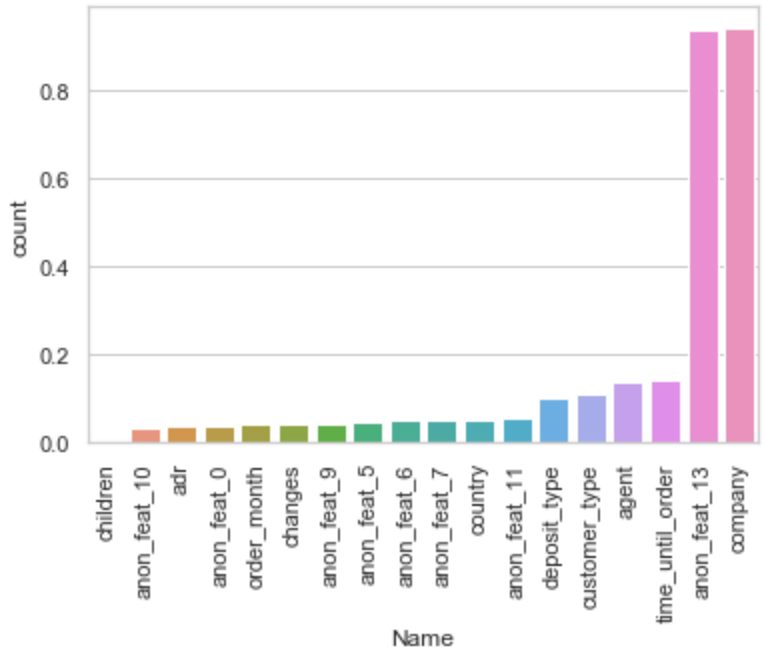
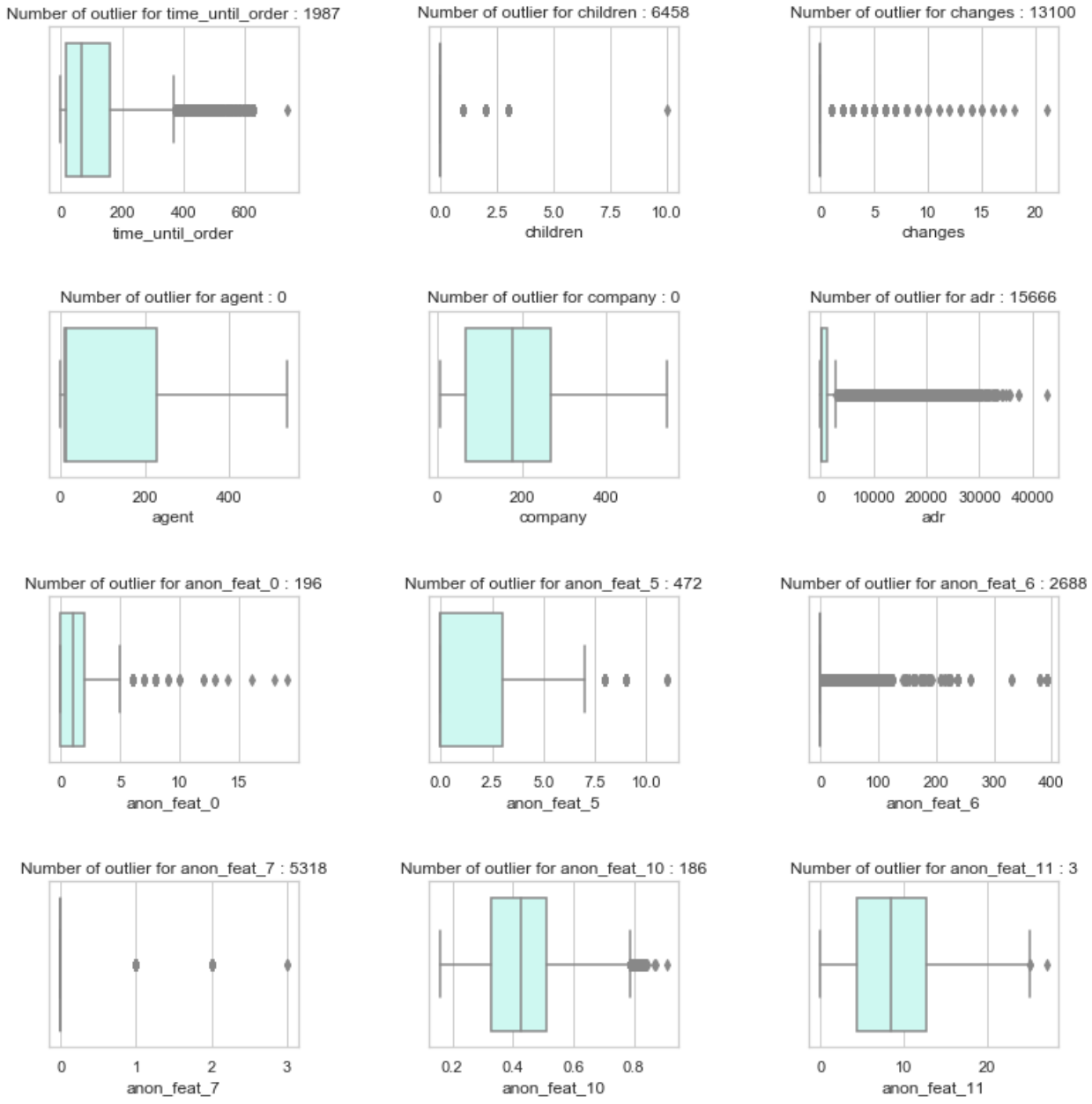
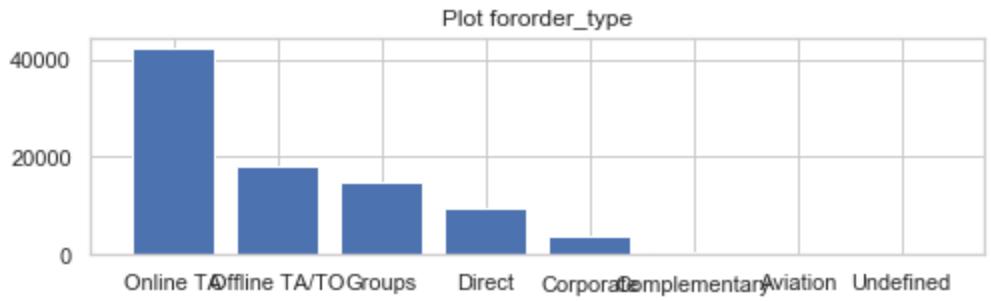
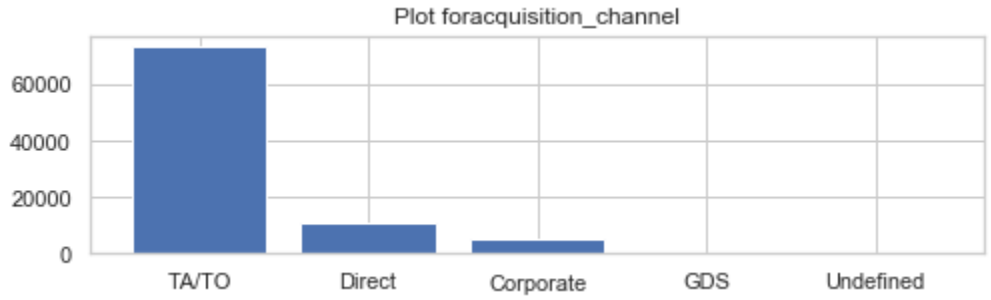


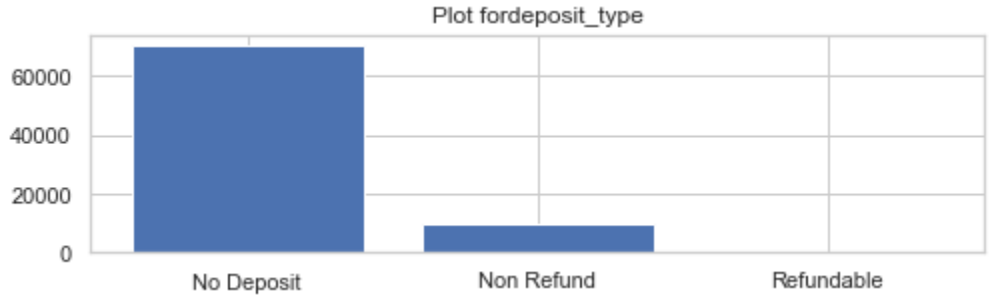
Figure 1.4 - Outliers detection with boxplots

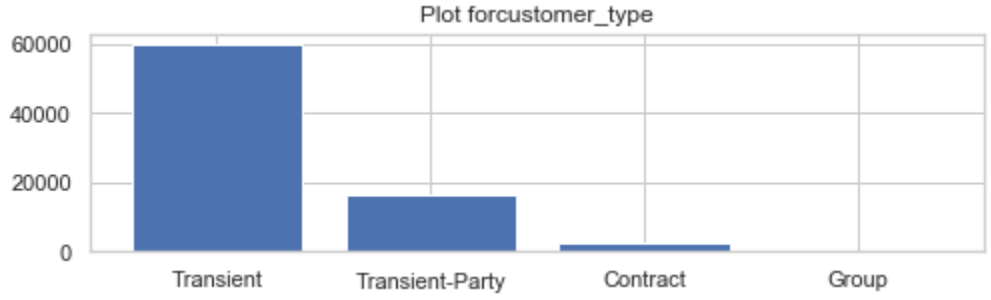


Figures 1.5-1.9 - Categorical outliers detection


Plot fororder_type
40000
20000
0
Online TA
Offline TA/TO
Groups
Direct
Corporate
Complementary
Aviation
Undefined


Plot foracquisition_channel
60000
40000
20000
0
TA/TO
Direct
Corporate
GDS
Undefined


Plot fordeposit_type
60000
40000
20000
0
No Deposit
Non Refund
Refundable


Plot forcustomer_type
60000
40000
20000
0
Transient
Transient-Party
Contract
Group

Chapter 2 - preprocessing

Table 2.1 - dropped features Due to feature selection

Dropping reason	Name of the feature
Creating gdp_rate as a new feature and a replacement for country features.	Country
94% of the data are None values.	Company
93% of the data are None values.	Anon_feat_13
99% of the data are zero values and was replaced by 'family' feature.	Babies
94% of the data are zero values and was replaced by 'new_customer' feature.	Prev_canceled
96% of the data are zero values and was replaced by 'new_customer' feature.	Prev_not_canceled
92% of the data are zero values and was replaced by 'family' feature.	Children
92% of the data are zero values.	Anon_feat_6
96% of the data are zero values.	Anon_feat_3
Almost all outliers	changes
Before creating this dummy variable we have identified it has a very low observation amount, marked as an outlier.	'order_type_Aviation', 'order_type_Complementary' ,'order_type_Undefined' ,'acquisition_channel_GDS', 'acquisition_channel_Undefined' ,'deposit_type_Group' ,'deposit_type_Refundable' ,'customer_type_Group'
After creating the dummy variables we have checked the zero values percent of the data, we have received for both features more than 95%.	order_type_Corporate, customer_type_Contract

Chapter 3 - Modeling

Table 3.1 - models chosen+default parameters

model	parameters
KNN	{'algorithm': 'auto', 'leaf_size': 1, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': None, 'n_neighbors': 26, 'p': 1, 'weights': 'uniform'}
Logistic regression	{'C': 1.0, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_scaling': 1, 'l1_ratio': None, 'max_iter': 100, 'multi_class': 'auto', 'n_jobs': None, 'penalty': 'l2', 'random_state': 6, 'solver': 'liblinear', 'tol': 0.0001, 'verbose': 0, 'warm_start': False}
Random forest	{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'entropy', 'max_depth': 50, 'max_features': 'sqrt', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_impurity_split': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 50, 'n_jobs': None, 'oob_score': False, 'random_state': 6, 'verbose': 0, 'warm_start': False}
MLP	{'activation': 'relu', 'alpha': 0.0001, 'batch_size': 'auto', 'beta_1': 0.9, 'beta_2': 0.999, 'early_stopping': False, 'epsilon': 1e-08, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'max_fun': 15000, 'max_iter': 200, 'momentum': 0.9, 'n_iter_no_change': 10, 'nesterovs_momentum': True, 'power_t': 0.5, 'random_state': 6, 'shuffle': True, 'solver': 'adam', 'tol': 0.0001, 'validation_fraction': 0.1, 'verbose': False, 'warm_start': False}

Chapter 4 - Model Evaluation Metrics

Figure 4.1 - Confusion Matrix of random forest model

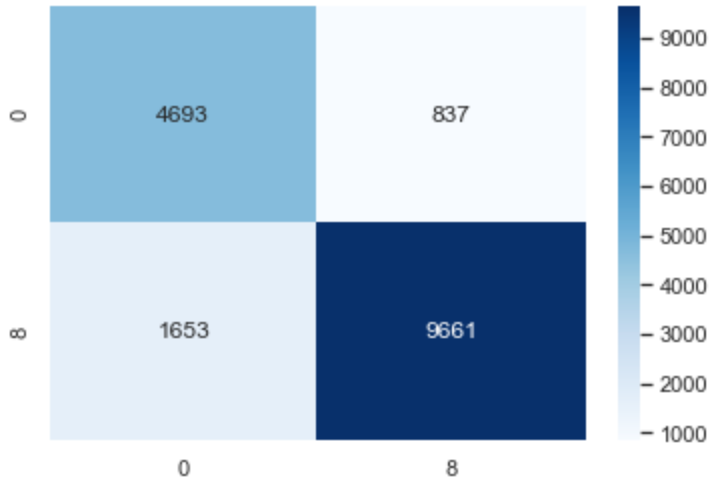
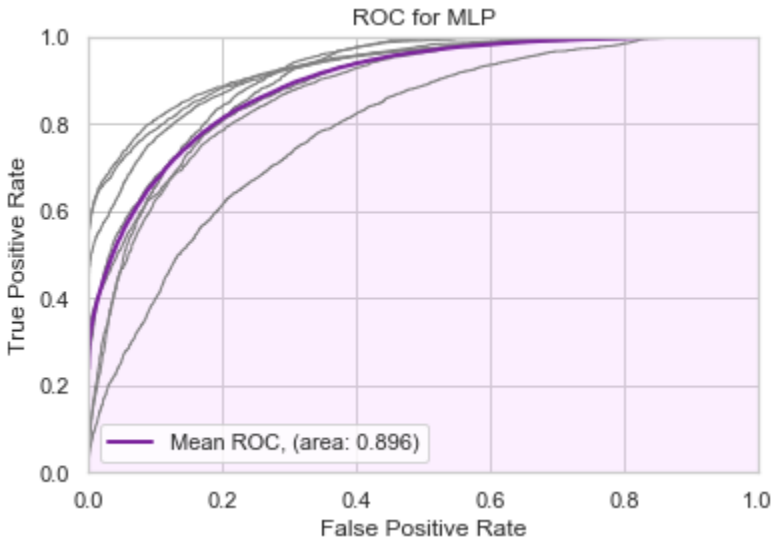
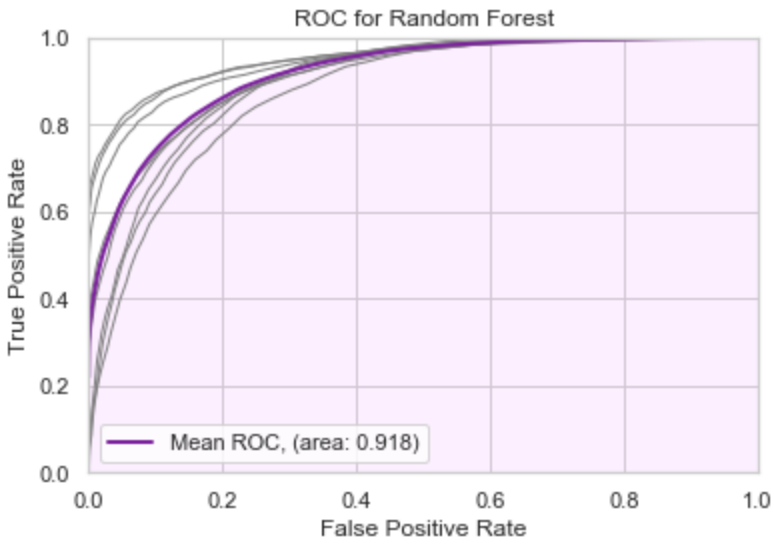
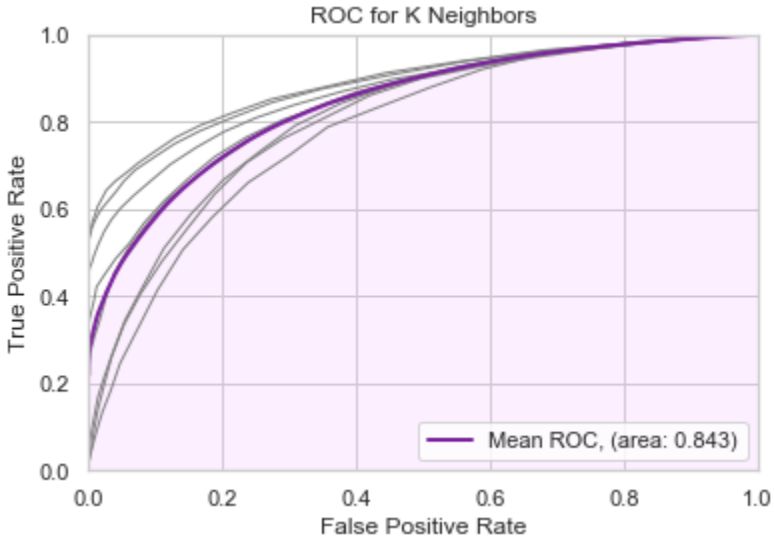
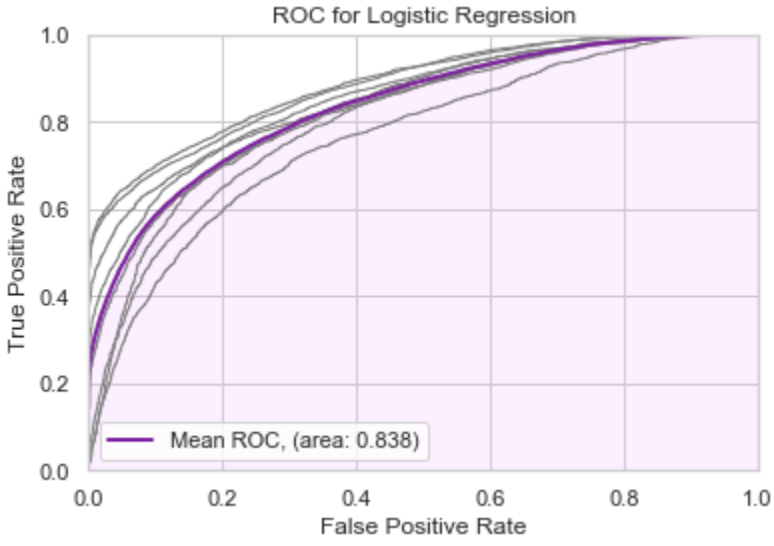


Figure 4.2 - ROC curve of K-fold cross validation




ROC for Random Forest
True Positive Rate
False Positive Rate
Mean ROC, (area: 0.918)
0.0
0.2
0.4
0.6
0.8
1.0


ROC for K Neighbors
True Positive Rate
False Positive Rate
Mean ROC, (area: 0.843)
0.0
0.2
0.4
0.6
0.8
1.0


ROC for Logistic Regression
True Positive Rate
False Positive Rate
Mean ROC, (area: 0.838)
0.0
0.2
0.4
0.6
0.8
1.0

Chapter 5 - functions documentation

Table X.1 - functions documentation

function	Input (parameters)	Description and output
boxplot_features()	df	Creating boxplot for each numerical column and showing the number of outliers on each title.
categories_outliers()	df	Showing number of observations for each sub-category in each categorical feature. The goal is to identify which sub-category is an outlier to be removed afterwards.
check_na()	df	Showing number of None values and percent for each feature.
check_zeros()	df	Showing number of None values and percent for each feature.
correlationPlot()	df	Plotting correlation plot with important manual settings.
new_customer()	prev,prev_not	Creating a new feature by prev_canceled and prev_not_canceled values, if both are equal to zero it means the observation is a new customer (1) else it will receive 0.
fill_none_mean()	df	Fill None values of anon - 0,5,6,7,9,10,11 with the mean value.
numeric_month()	df	Mapping the name of the month to the number of it.
month_to_season()	Month - number of month	Creating a new feature which shows the name of the Season based on the name of the month.assuming that the seasons around the world are uniform.
numeric_order_week()	Val - value for each	Cutting the prefix “week_”, leaving only the

	observation.	number of week as an integer.
add_gdp()	df	creating new feature which based on external countries data. by merging by the country name we will it is gets the GDP of the every country
changeNoneToMean()	df	Changing None values for ‘time_until_order’ feature to mean value.
fill_outliers_mean()	df	Changing the outliers for time_until_order to mean value.
get_outliers()	df,col	Calculating the outliers by quantile 0.25 and 0.75.
categorical_nan_handler()	df	Changing None values for all categorical feature to a new sub-category ‘other’
count_zeros()	df	Calculating amount of zero and returning a list of features with more than 95 percent of the data.
drop_features()	df,feature_list	Removing features from the data by the feature_list valus.
preprocessing()	df	Getting a dataframe. Based on all other functions which have declared it arranges the data to final data to work with.
run_model()	X_train_scaled, X_valid_scaled ,y_train,y_test, clf	Running a model by the clf that was received and returning the data needed for the confusion matrix.
plot_confusion_matrix	auc,cm	Plotting a confusion matrix by the given values.
print_stats()	y_test,y_pred,clf,X_test,X_train,y_train	Printing summary of Important metrics such as: accuracy, precision, recall, AUC.
grid()	X_train, X_test	Running a grid search by the clf and parameters that was given, return the best

	,y_train, Y_test ,clf ,params	parameters that were found.
cancelation_bars()	column_name	Plotting a bar plot which describes how many true/false cancelations per category of categorical feature.
importance()	df,labels,clf	Getting features and labels. By applying the feature importance function of Random Forest classifier, it prints a summary of the importance and plots the results.
KfoldPlot()	X,y,clf,k	Getting a data frame, labels and classifier, it is iterating over different folds of specific k value. Returns ROC plot with the different AUC results of different folds and the mean of them.
final_model()	X_train_scaled ,X_valid_scaled ,y_train, Y_test ,clf	For the best model that we chose, the function will return the test-auc and the train-auc.