**Data Science Project Protocol**

*Author(s):*

*Jenny Shuster*

# Introduction

This project is about the no-show problem that lots of businesses have to deal with.

Particularly, in this challenge the focus is on the no-show cases of a hair salon.

No-shows are a big problem for hair salons the same as airlines and medical facilities.

The question we would like to find the answer for: Which booking are probably going to be canceled?

What do we already know about the problem?

Excessive no-shows increase costs and wait times for businesses and all other customers alike. A No-show prediction service would allow hair salons to select from a variety of treatment options at the time of the booking such as:

1. Requiring a non-refundable deposit.
2. Scheduling the appointment at a different time, location.
3. Scheduling the appointment with a different service provider such that the potential no-show would have less business and customer experience impact.

The outcome of the chosen algorithm will be the probability of a booking to be canceled.

The hair salon can decide which of the errors is more important to avoid: False positive or False negative, and choose the optimal threshold for his business.

What is known to influence the outcome?

The most significant factor which influences the no-show rate, as written in professional articles, is the **customer experience**.

How can we measure it? Well, we should ask if the client was happy with the following:

1. Their results last visit – in our data, the staff personal which delivered the service last time has a big impact on the next booking to be canceled: his/her attitude, his/her experience.
2. The value for money – the price of the service.

Another known significant factor is the specific customer. Some customers just tend to cancel their appointments more often than others. Hair salons try to avoid the damage by warning the client, and sometimes eventually end up adding those clients to a blacklist.

Since the data set does not include any personal information about the client, we can try to identify a cancellation pattern of the clients by recognizing past bookings and cancellations of each client id code.

Do we have any possible new knowledge that has not been in use before?

The dataset did not include any information about booking hours, which is mentioned in some professional forums as a possible factor. Another factor mentioned in recent researches: the number of days that have passed since the last booking. The latter can be calculated.

# Methodology (Project design)

## Data

Which data will be used?

Our data sources are:

* Client Cancellations0.csv - Client Cancellations are bookings that have been canceled.

This file has mixed types of data in the date columns.

We will fix it via python and use the fixed version: ClientsCancellation1.csv.

* Future Bookings (All Clients)0.csv - Future bookings is the set of all uncanceled bookings.
* No-Show Report0.csv - A listing of no-show bookings that were not canceled prior to the booking date, the client simply did not show up. We will treat each one of those as a cancelation at the day of the booking.

This file has mixed types of data in the date columns.

We will fix it via python and use the fixed version: NoShowReport1.csv.

* Product Listing (Retail)0.csv - This is a listing of all retail products.
* Receipt Transactions0.csv - This is a list of all of the transactions with receipts.

This file has mixed types of data in the date columns.

We will fix it via python and use the fixed version: ReceiptTransactions1.csv.

* Service Listing0.csv - This is a listing of all services.

The inclusion/ exclusion criteria:

The set of all bookings is the union of ‘Client Cancellations1.csv’, ‘Future Bookings (All Clients)0.csv’ and ‘NoShowReport1.csv’.

Using all the sources, we will create a data set, where each row is a booking.

The criteria for a valid row that can be included as a booking:

* Date
* Staff code
* Client code
* Service code
* 1 for a no-show and 0 for a booking which was not canceled.

Each unique combination of these is a booking.

The time frames periods:

As we mentioned, a booking is a unique combination of Date, Staff code, Client code ,Service code, and 1 for a no-show / 0 for a booking which was not canceled.

This means that we can have more than one booking in the same day for each client.

The train and test will be divided randomly.

* The problem is of classification kind. The outcome variable is “IsCanceled” which is 0 for the uncanceled bookings from “Future Bookings (All Clients)0.csv”, and 1 for the canceled bookings in “ClientsCancellation1.csv” or “NoShowReport1.csv”.
* Possible sources of leakage that can create a bias and will be dropped:
  + "Days" – number of days between the cancelation date and the booking date.
  + "CancelDay" – the day of cancelation. Reveals that the booking was canceled.
  + "CancelWeekday" – the week day of cancelation. Reveals that the booking was canceled.
  + "CanceledBy" – the staff personal that technically canceled the booking. Reveals that the booking was canceled.
* In order to decide which additional variables we can add, we will create a No-Show database in our SQL server with a table for each csv source and query those tables.

It will help us understand the way some tables can be joined, discover unique values in some columns and more.

* To enrich the data, we can apply techniques such as selecting the data of the last booking of the client, counting number of past bookings for the client, counting past cancelations for each client or staff personal at each booking and so on. All of those can be done using SQL techniques like: window functions, joins, CTEs and more.  
  Finally, a VIEW in the database will be our flat file.
* The next step is exploring the data of this flat file via python, using EDA techniques:
  + Looking at the min, max, mean and median values.
  + Looking at the proportions of the “IsCanceled” values to understand whether the data is balanced or not.
  + Counting missing values in each variable.
  + Visualizing the distribution of values in each variable.
  + Calculating and visualizing the correlation matrix.
* Dealing with the outliers

They will be explored using IQR based Extreme Value Analysis, since in the EDA we discovered that we are dealing with a non-parametric data. Then, using significance test, we will find out if the outliers affect the assumptions.

For the variables whose outliers affect the assumptions, we will check if they also change the results using distribution visualization of each variable for each “IsCanceled” category.  
For those variables whose outliers change the assumptions, we can replace the values with NAN (Null), but before doing that we should take into account the existing missing values in those variables, and their missing mechanism.

* Dealing with the missing values

In the EDA, we will calculate the missing percentage of each variable, and divide the variables into 2 categories:

1. Columns with too high missing percentage that need to be dropped.
2. Columns with high missing percentage that can be categorized

If there will be columns with a low percentage of missing values, we can drop those rows.

* More information can be found in the Data retrieval protocol.

## Models

* With our clean data, we will perform our feature selection.  
  For the majority of the variables, which are now categorical, we will use Chi-square test.  
  The t-test which took place in the EDA can help us with the continuous variable 'ServicePrice'.
* Then, the dataset will be divided to 3 parts of training, validation and test using the following technique:
  + Using the function “train\_test\_split” (Python), we divide it randomly, to 20% test and 80% train and validation.
  + In each model, in the training phase we will divide the 80% to 40% train and 40% validation as a part of the cross-validation technique.
* Models we will train and validate:
  + Logistic Regression Classifier
  + Random Forest Classifier
  + XG Boost Classifier
  + SVM Classifier
* The outcome - each model will return a probability for a booking to be a no-show.
* The measure will be AUC, since we are dealing with a classification task and the data is not balanced: we have much less No-Shows than regular bookings (where the client did show up).   
  Using the mean AUC of the validation, we will see what the best models are, and also evaluate the optimal parameters in the model fine-tuning.
* In our case, the best models were Random Forest Classifier and XG Boost Classifier, with a very small difference between them.  
  Random Forest Classifier was slightly better, but XG Boost Classifier has an important advantage in its built-in logic:
  + It uses random sample with replacement over weighted data - higher weights are given to misclassified data to emphasize the most difficult cases. Those cases will be more likely to be selected on the sampling phase, so subsequent models could focus on them.

For all those reasons, we will use an ensemble of Random Forest Classifier and XG Boost Classifier.

## Deployment of your model

* A QA and a data analyst will check the project, using the following protocol:
  + Check for insert errors – compare the csv to the tables in the DB.
  + Check for errors in the queries of the bookings VIEW and the flat-file VIEW.
  + Check for leakage variables.
  + Check for missing values in each variable.
  + Check for multi correlations between variables.
  + Check for random division of the data (train, validation, test).
  + Check the results of the validation – are the chosen models really the best ones?
* The final user of the prediction is the staff of the hair salon.  
  After a business meeting with the hair salon manager, a threshold will be chosen, thus the model will be able to retrieve a prediction: is this booking likely to be a no-show?
* The prediction will be presented to the final user – the staff member on their computer.

When a new booking is inserted: Client code, Service code, Staff code and date, all the features will be calculated and the model will calculate the prediction.

The hair-salon will receive a notification via email and the staff would be able to decide about the next steps.

* The final user will be able to interpret the prediction easily, using the information that will be sent in the mentioned email.
* The model will run on our server, and provide the results.
* The model will be updated each month.
* If the model will receive an incomplete data, it will send an email with the missing data out of the 4 mentioned: Client code, Service code, Staff code and Date.
* For the final prediction, we will use an ensemble of Random Forest Classifier and XG Boost Classifier.
* The measure AUC was used to evaluate the prediction.
* For the test data we had, the mean AUC was for ensemble of Random Forest Classifier and XG Boost Classifier: 0.7976

# Results

Here you will present the main results of all the process. We will describe:

* The final amount of data used: 2764 bookings. Out of them -
  + 553 for test
  + 1106 for train
  + 1105 for validation
* Outliers:   
  It turns out we do not have outliers that affect both the assumptions and the results.  
  The variables whose outliers affect the assumptions but not the results:

"DaysRecency","TotalClientCancellationsTillNow","TotalClientBookingsTillNow",

"CancellationPcnt", "ServiceCost".

* + "DaysRecency", "TotalClientCancellationsTillNow" - already contains nearly 30% of NULLs which represent new clients. It will be categorized in the Missingness treatment part.
  + "CancellationPcnt" – its normal values are between 0 and 1. We can see the IQR based Extreme Value Analysis tagged values >= 0.65 as outliers. Seems like we need to use a higher iqr factor for this particular variable to detect outliers. When we change it, we can see the outliers disappear.

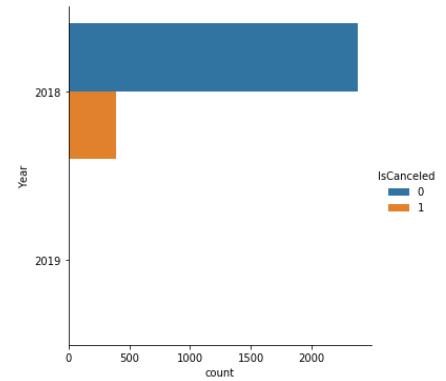
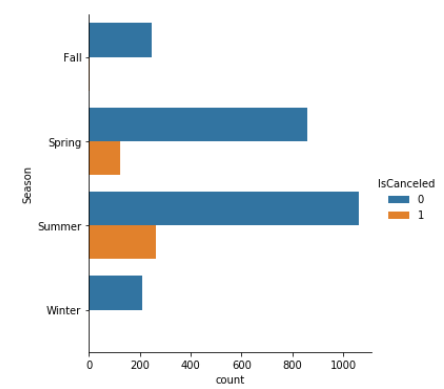
For all those reasons, we will treat now only outliers of “TotalClientBookingsTillNow", "ServiceCost".

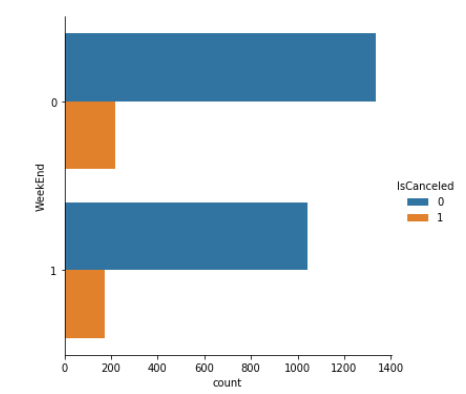
* Missing values:

In the EDA, we calculated the missing percentage of each variable, and divided the variables into 2 categories:

1. Columns with very high missing percentage = Over 70% missing.  
   Treatment: Drop the variable.
2. Columns with high missing percentage = 30% missing (same percent for all) - All those variables have MNAR type of missing: if there were no visits in the hair salon for this client, in other words – if this is a new client, all those variables are NULL-> Treatment: Categorization.

* The distribution of the data (timeframes):





Additional visualizations of the distributions can be found in the jupyter notebook.

* The methods used to transform the data and to generate new features were SQL (techniques like: window functions, joins, CTEs and more) in the first phase of creating the flat file.  
  In the following parts of the project, we used Python (numpy, pandas, scipy.stats functions, as well as functions we defined ourselves).

# Conclusion

The project began as my first data science project. The goal was to create a machine learning pipeline which will be able to help in detecting future no-shows of a hair salon.

The most important challenges I had:

* The original csv files contained date format problems which had to be solved, otherwise – it was impossible to work with it.
* The data set did not include some important variables, such as:
  + Hours of bookings.  
    The time of the day could be a good feature to predict no-shows.
  + Client information, besides his/her unique identification code.

Different characteristics of the client could lead to more interesting and general conclusions, and may also lead to a better prediction.

For those reasons, it was very important to be creative and think of lots of new features, which are based on relatively small variety of existing columns.

* The complex calculations of those new features, demanded a deep, careful check of the SQL code.
* There were some challenging decisions that had to be taken, regarding variables with outliers and missing values – as a result of identifying the missing mechanism as MNAR, some of them had to get the missing treatment prior the outlier treatment.
* In the correlation matrix from the EDA phase it was clear that there is a high correlation between "TotalClientCancellationsTillNow","TotalClientBookingsTillNow" and "CancellationPcnt"."CancellationPcnt", "TotalClientCancellationsTillNow" have the higher correlation with our target, so it seemed natural to assume that in the feature selection phase we will have to drop the "TotalClientBookingsTillNow".  
  Eventually, while optimizing the alpha of the Chi-squre test we used for the feature selection – it was discovered that the use of all three "TotalClientCancellationsTillNow","TotalClientBookingsTillNow" and "CancellationPcnt" give us better results.

The final prediction was achieved by:

* Optimizing the alpha of the Chi-square test we used for the feature selection – to be 0.5.
* Optimizing the Cross-Validation value – to be 2.
* Optimizing the hyper-parameters using the grid-search function.
* The final model is an ensemble of the Random Forest Classifier results multiplied by the weight of 0.1 and XG Boost Classifier results multiplied by the weight of 0.9.

Model limitations:

* The model will work only when it will receive non NA data for the date, client code, staff code and service code of the booking. Since there were no NAs for these features, we assume this will not be an issue.
* There is a threshold decision that needs to be taken. The hair-salon must choose what is more important to avoid: False-Negatives, or False positives? There is a tradeoff.
* Since there was no data about the general characteristics of the clients or staff, and also the hair salon (location in the city, for example), the model might be good enough only for this specific hair salon.