# Introduction

Variable Information:

• “Transaction” is the target variable. FALSE where no transaction occurred during a session, and TRUE where a transaction occurred.

• "Administrative", "Administrative\_Duration", "Informational", "Informational\_Duration", "ProductRelated" and "ProductRelated\_Duration" represent the number of different types of pages visited in a session and total time spent in each of these page categories.

• "Bounce Rate", "Exit Rate" and "Page Value" represent the metrics measured by "Google Analytics". The value of "Bounce Rate" refers to the percentage of visitors who entered the site from a specific page (landing page for a session) and then left ("bounced") without visiting any other page during that session. The value of "Exit Rate" is calculated as for all pageviews to a specific page (exit page for a session), the percentage that were the last in that session. The "Page Value" represents the average value for a web page that a user visited before landing on the goal page or completing an e-commerce transaction (or both) in the given session. Goal page is a page that e-commerce company wants visitors to reach during a session. This page can be a transaction page. Shopping cart pages often have high page values.

• "Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother’s Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction. The value of this attribute is determined by considering the dynamics of ecommerce such as the duration between the order date and delivery date. For example, for Valentine’s day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8. • “Month” indicates the month when session took place.

• “VisitorType” indicates the type of visitor for a session – Returning\_Visitor or New\_Visitor

• “Weekend” is FALSE where session did not occur on weekend, and TRUE where session occurred on weekend.

# Part 1: Random Forest

Random Forest is....

## Data Preparation

The data preparation started by analyzing each variable from the provided dataset E-shop.csv. Firstly it was noticed that there was no missing values, as all the variables had to be numerical, each datatype was identified as well (figure 1). 4 out of the 14 columns weren’t numerical: 2 objects and 2 booleans.



Figure 1 - print of dataset.info

By analyzing the raw data the column “VisitorType” was identified as Boolean because it only had two available values: “Returning\_Visitor” or “New\_Visitor”. For this reason all the Boolean columns were converted using the “converter” function (figure 2), considering “New\_Visitor” and “True” as 1’s and “Returning\_Visitors” and “False” as 0’s.



Figure 2 - Converter Function

The remaining object column “Month” was converted using “pd.get dummies” creating a column for each month in our dataset, containing 1 wherever the initial column name was True and 0’s elsewhere.   
 Transaction is the target variable so we proceeded with the division of the dataset into feature and label sets. Therefore, with the dataset labeled and all variables being numeric, the data was ready to be normalized using the “StandardScalar()” method imported from sklearn, transforming each column to have a mean of 0 and variance of 1, ending the preparation of the data.

## Model Evaluation Strategy

After the data preparation, evaluating the model strategy follows. The goal is to predict whether transaction will take place during a given web session or not. For better clarification an analysis on the dataset content was required. By running the panda function “pd.Series(Y).value\_counts())” we are able to see if our data is balanced or unbalanced (figure 3). In this case there are over than 5 times more no-transactions than transactions on the visits of the website.



Figure 3- Balance analysis of our dataset

As our dataset is unbalanced we can’t focus only on the clarification accuracy of our model. Instead the focus has to be on the **reduction** of **False Positives** because the model needs to improve the capacity of predict who is going to buy, to for example be able to address better marketing to who doesn’t buy. So, getting false positive means that we predicted that they would buy and they didn’t, in this case it is better to predict that they didn’t buy.

## Model Building and Testing

To build the Random Forest model the data was randomly split to training data (70%) and the remaining (30%) to testing data, in order to validate the same.

As we have seen on section 2 our data is imbalanced. A balance training set ensures a balanced learning, for this reason a SMOTE (Synthetic Minority Oversampling Technique) was used on the training dataset.

For the Random Forest Model, the “RandomForestClassifier” function is going to be used with a criterion of “entropy” to gain information with the splits, a no “max\_depth” was used to determine the size of the trees, meaning that the nodes are expandable until all leaves are pure.

To find number for our model we need to tune the random forest parameter, to achieve this the “GridSearchCV” function from “sklearn.model\_selection” was used, using a 5-fold cross validation and a scoring parameter of “precision” in order to minimize the false positives. The list used as parameters for the grids estimators was: 50, 100, 150, 200, 250, 300. The method described lead to a tuning parameter of 150 as the best estimator of the number of decision trees, resulting in a mean cross-validated score of 0.912 (figure).



Figure 4 - Tuning the random forest paremeter "n\_estimators"

By running the random forest model with 150 decision trees it was found the significance of the variables (figure 5). The confusion matrix was generated as well with the following results:

* Total Positives: 418;
* Total Negatives: 2807;
* False Positives: 298;
* False Negatives: 151.

Considering the purpose of the prediction model the number of False Positives is very high.

|  |  |
| --- | --- |
| Variable | Significance |
| PageValue | 0.380946 |
| ExitRate | 0.086323 |
| ProductRelated\_Duration | 0.082307 |
| Administrative | 0.080639 |
| ProductRelated | 0.078779 |
| Administrative\_Duration | 0.064977 |
| BounceRate | 0.059778 |
| Month\_Nov | 0.034286 |
| Informational | 0.028492 |
| Informational\_Duration | 0.021505 |
| Month\_May | 0.015057 |
| VisitorType | 0.013911 |
| Month\_Mar | 0.010882 |
| Weekend | 0.010691 |
| Month\_Dec | 0.006601 |
| SpecialDay | 0.00604 |
| Month\_Sep | 0.004796 |
| Month\_Oct | 0.004147 |
| Month\_Jul | 0.004023 |
| Month\_Aug | 0.003559 |
| Month\_June | 0.001702 |
| Month\_Feb | 0.00056 |

Figure 5 - Significance of variables achieved with Random Forest with 150 decision trees

## Identifying Best Model

In order to identify the best model we need to select a number of top significant features and re-run our Random Forest with tuning parameter of 150 decision trees, we will be varying the number of features to try and reduce the number of False Positives and avoid overfitting of the model.

Starting with the top 5 features (“PageValue”, “ExitRate”, “ProductRelated\_Duration”, “Administrative”, “ProductRelated”) the model was re-run and achieved the following results:

* Total Positives: 403;
* Total Negatives: 2788;
* False Positives: 317;
* False Negatives: 166.

We have increased the number of False Positives, so we will try now with 6 (adding “Administrative\_Duration”) giving a number of false positives of 330. Adding another feature (“BounceRate”) we reduced to 312. More variables were kept being adding until the least number of False Positives was achieved (afterwards the value increased), with 12 variables, giving a number of 292 (figure 6), the best model was identified.

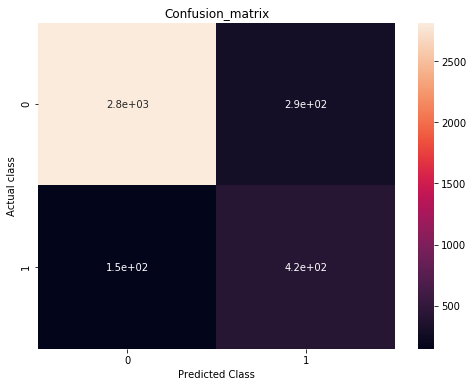


Figure 6 - Confusion Matrix of the best Random Forest Model

The features used to achieve the best model were “PageValue”, “ExitRate”, “ProductRelated\_Duration”, “Administrative”, “ProductRelated”, “Administrative\_Duration”, “BounceRate”,”Month\_Nov”,”Informational”,”Informational\_Duration”,”Month\_May”,”VisitorType”. Its performance is not good as it is doing a bad job, giving a high number of users buying that actually didn’t have a transaction. The model shouldn’t be used to predict the transactions of this shop.

## Generating Recommendations

# Part 2: PCA & K-Means Clustering

## PCA Implementation to visualize dataset

## Elbow Plot Creation

## K-Means Implementation