Tools for data analytics

**B8IT106**

**Team members:**

1. **Romina fratoni**
2. **Gregory Cannella**
3. **Shakir Aleem**

**Teacher:**

**Kunwar Madan**

**Submit Date: 27/04/2020**

Table of Contents

[Part 1 (Random forest) 3](#_Toc38707161)

[1 – Data preparation 3](#_Toc38707162)

[2 – Model Evaluation Strategy 4](#_Toc38707163)

[3 – Model building and testing 4](#_Toc38707164)

[5 – Generating Recommendations 7](#_Toc38707165)

[Part 2 (PCA): 8](#_Toc38707166)

[Analytsis: 8](#_Toc38707167)

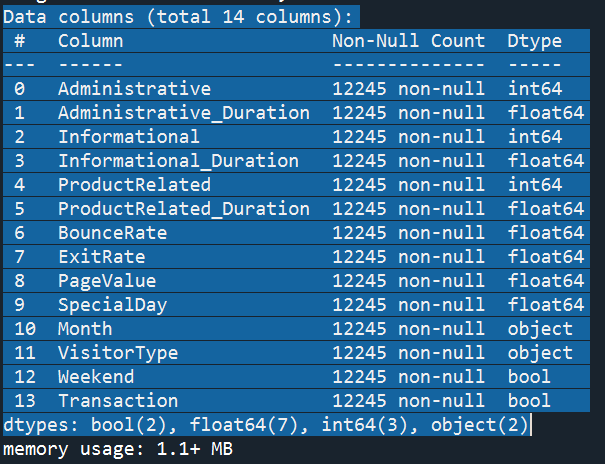
[Part 2(K-mean cluster): 8](#_Toc38707168)

[Analysis: 8](#_Toc38707169)

# Part 1 (Random forest)

## 1 – Data preparation

We familiarised ourselves with the different variables and identified our target variable to be the “Transaction” column.



Running the dataset.info() command (see picture above), we could confirm that the dataset has no missing values and that the data types are identical within each column. All columns are relevant to training our model and predict target column. Therefore, no column is dropped from dataset.

There are “VisitorType” and “Weekend” columns which have non-numeric and non-Boolean values. Therefore, we converted categorical “VisitorType” and “Weekend” columns into categorical feature columns.

We split our dataset into a training set (70% of the records) and a test set (30% of the records), so that we can train our model on the training set and then test our model on the test set. We also normalised our numerical features so that each feature has a mean of 0 and variance of 1.

Our target and independent value columns have unbalanced data. So, we have also balanced our imbalanced dataset, we used the Synthetic Minority Oversampling Technic (SMOTE) on our training set to increase the number of records in the minority class (i.e: transaction takes place).

## 2 – Model Evaluation Strategy

The goal of creating prediction model is to increase number of transactions on website. Model will help identify which individual columns contributing more to complete transaction process. Also model will assist to identify number of provided data are relevant enough to create random forest predictive model.

What we are trying to achieve here is to correctly predict when a transaction takes place. In other words, we are trying to build a model that can correctly predict true positives.

The assumption is made in this regard that if minimise wrong prediction of complete transaction i.e. minimise false positive then it will increase the numbers of wrong prediction of transaction which not happen.

The decision has been made assuming some Business scenarios too. The impact of having higher value of false positives could be more dangerous than having higher values of false negatives because:

1. The company, through the number of predicted transactions, could decide to increase the stock of certain products in order to prevent running out-of-stock, delays and customers dissatisfaction.
2. The company, through the number of predicted transactions, could decide to make investment in technology to get more performant website and avoid more abandoned sessions.
3. Company also wants to improve its day to day operation like logistic, delivery of orders, work force working in warehouse etc.

These decisions could lead a huge cost increase driven by the forecasted successful transactions. Minimizing the false positives could minimize the risk to invest in higher costs than needed.

Model accuracy has not been a good indicator here, since we are dealing with a very much imbalanced dataset.

## 3 – Model building and testing

In order to find the correct number of decision trees we built our Random Forest model and we used the Grid Search algorithm using a list of n\_estimators in a range from 50 to 400 with an interval of 50. We also passed in “precision” in our scoring parameter in order to minimise false positives.

Since our result came at the middle boundary (250), we then set estimator to 250 to decision tree.

The number 250 ended up being the best parameter to use for our Random Forest model.

Please find below the performance for the model using all features and for the model using a subset of the features.

**Mean cross Value:**

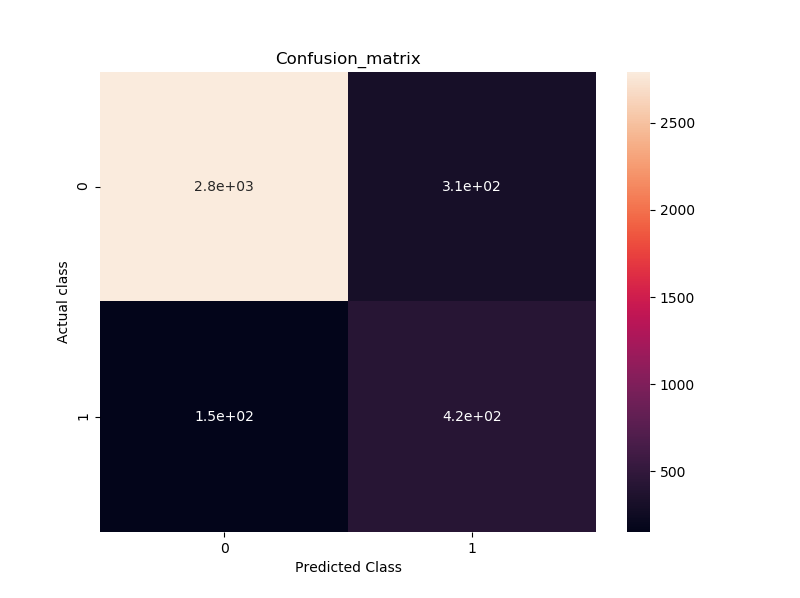
The mean cross validated score is 250 estimator is 0.9118422167709991 i.e. 91%.

Average percentage of variation for the target variable that is explained by each feature:

|  |  |
| --- | --- |
| PageValue | 0.372773 |
| ExitRate | 0.086930 |
| ProductRelated\_Duration | 0.083587 |
| Administrative | 0.082638 |
| ProductRelated | 0.077507 |
| Administrative\_Duration | 0.066712 |
| BounceRate | 0.062556 |
| Month\_Nov | 0.032657 |
| Informational | 0.027552 |
| Informational\_Duration | 0.022183 |
| Month\_May | 0.015238 |
| Weekend | 0.010876 |
| Month\_Mar | 0.010414 |
| VisitorType\_Returning\_Visitor | 0.008499 |
| VisitorType\_New\_Visitor | 0.007804 |
| Month\_Dec | 0.006607 |
| SpecialDay | 0.005988 |
| Month\_Sep | 0.005115 |
| Month\_Oct | 0.004528 |
| Month\_Jul | 0.003846 |
| Month\_Aug | 0.003643 |
| Month\_June | 0.001810 |
| Month\_Feb | 0.000539 |

**Performance using all features:**

|  |  |
| --- | --- |
| TP | 419 |
| TN | 2793 |
| FP | 312 |
| FN | 150 |

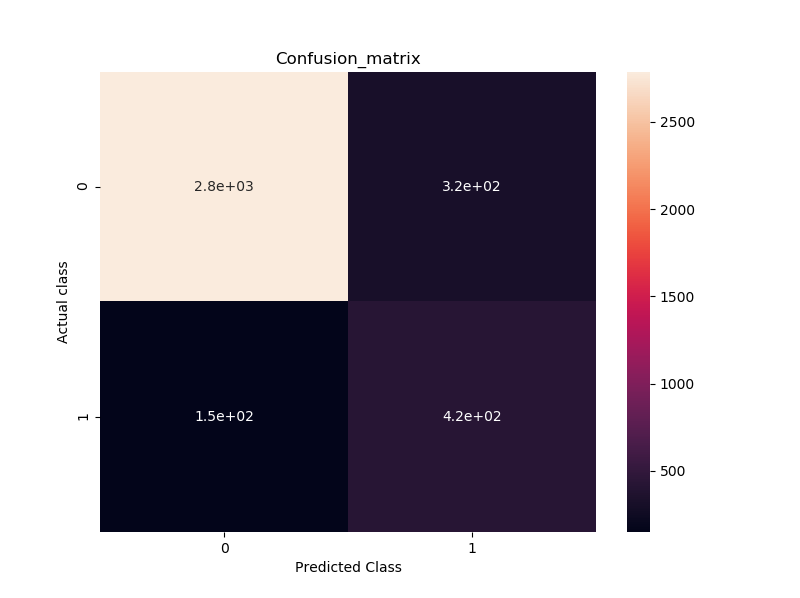


**Performance using a subset of features:**

List of features used in correct significance order: Page Value, ExitRate, ProductRelated\_Duration, ProductRelated, Administrative, BounceRate, Administrative\_Duration, Month\_Nov, Informational, Informational\_Duration, Month\_May, Weekend

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| number of features | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| True Positives | 345 | 414 | 410 | 401 | 412 | 414 | 420 | 415 | 415 | 415 | 415 | 422 |
| False Negatives | 224 | 155 | 159 | 168 | 157 | 155 | 149 | 154 | 154 | 154 | 154 | 147 |
| percentage of correct prediction | 0,61 | 0,73 | 0,72 | 0,70 | 0,72 | 0,73 | 0,74 | 0,73 | 0,73 | 0,73 | 0,73 | 0,74 |

|  |  |
| --- | --- |
| TP | 420 |
| TN | 2786 |
| FP | 319 |
| FN | 149 |



Based on the above figures the total number of “Real” positives values in our test dataset is: True Positives + False Negatives = Real positives

420 + 149 = 569

Percentage of correct prediction of “Real” positives = 420 / 569 = 0.7381

We can conclude that our model was able to correctly predict 73.81 % of the “Real” positives in our test dataset.

## 5 – Generating Recommendations

From the analysis made on the previous point we can assume that the best final model is built using the first feature as relevant.

As the first feature ‘PageValue’ is a measure of how ‘good’ the page is in order to drive the customer to make the transaction, we could suggest:

1. Identifying the pages with higher ‘PageValue’ metric and analyse them in order to understand the behaviour of customer while visiting the page;
2. Build sales funnel analysis in order to identify the events on the page that cause the most part of loosing customers before they finalise the transaction;
3. Utilising Web Analytics tools to identify the less valuable elements of the page that bring the customers to lose focus on making transaction (i.e. external links that pick attention and distract the customer)

# Part 2 (PCA):

PCA also implemented on the created dataset which was used in random forest analysis. To visualize all independent variables on 2D chart we used PCS.

We will analyse how much variance 1st two components will explain so that we can trust our visualization or not. Number of components set to 2. We passed our normalised dataset into PCA model. This will give us the values 1st and 2nd principle components.

## Analytsis:

The result of PCA function show as following

|  |  |
| --- | --- |
| Component | Result |
| 1st principal component | 0.26485881 (26%) |
| 2nd principal component | 0.12899158 (12-13%) |
| Sum of 1st and 2nd components | 0.39385039480465966 (~40%) |

The above result shows that the visualisation result of our normalised dataset is very poor. The result must be above than 70%.

Hence PCA model showing only around 40% of variance of data which mean PCA has failed to visualized variance. We cannot trust this visualization.

# Part 2(K-mean cluster):

Like PCA we are going to use two components to implement K-means clustering. We pass the PCA X and Y component to k-mean. K-means will identify the cluster and label them rather than creating them.

## Analysis:

As PCA is not showing variance more than 40%, we can’t not trust k-mean clustering result.

**Elbow plot:**

To guess the number of cluster in dataset we use Elbow plot. From the Elbow plot we can see the number of cluster is 3. This shows that the actual number of cluster could around 3 i.e. 2 or 4.