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)

1 – Data preparation

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

```
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12245 non-null  int64
 1   Administrative_Duration  12245 non-null  float64
 2   Informational            12245 non-null  int64
 3   Informational_Duration   12245 non-null  float64
 4   ProductRelated           12245 non-null  int64
 5   ProductRelated_Duration  12245 non-null  float64
 6   BounceRate               12245 non-null  float64
 7   ExitRate                 12245 non-null  float64
 8   PageValue                12245 non-null  float64
 9   SpecialDay               12245 non-null  float64
 10  Month                    12245 non-null  object
 11  VisitorType              12245 non-null  object
 12  Weekend                  12245 non-null  bool
 13  Transaction              12245 non-null  bool
dtypes: bool(2), float64(7), int64(3), object(2)
memory usage: 1.1+ MB
```

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 be included in our training dataset to predict the target column. Therefore, no column is dropped from dataset.

The "VisitorType" and "Weekend" columns have non-numeric and non-Boolean values. Therefore, we converted the categorical "VisitorType" and "Weekend" columns into numerical values using the get_dumies() function.

The reason we used the get_dumies() function is to split the different values into their own columns in order to see the impact or variation explained by each of them.

We then 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.

Because we are dealing with a very much 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 the prediction model is to increase the number of transactions on the company's website. The model will help predict which web sessions will likely lead to transactions and identify the most relevant features.

Model accuracy would not have been a good indicator here, since we are dealing with a very much imbalanced dataset. For our target variable, we roughly have 85% of the records being "False" and 15% of the records being "True".

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.

We therefore want to minimise false positives in order to increase the prediction accuracy of our model in correctly identifying True Positives. By minimising False Positives, we understand that it will increase the number of False Negatives.

The decision has been made assuming some Business scenarios too. The impact of having higher number of False Positives could have a more negative impact rather than having a higher number 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, causing 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 to reduce the number of visitors leaving the website without making a transaction.
3. The company could also change or improve its day to day operations like logistics, order preparation, resource management, etc...

These decisions could lead to 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.

3 – Model building and testing

In order to find the correct number of decision trees, we built our Random Forest model using the Grid Search algorithm with a list of n_estimators going from 50 to 400 with an interval of 50. We also passed in "precision" as our scoring parameter in order to minimise false positives.

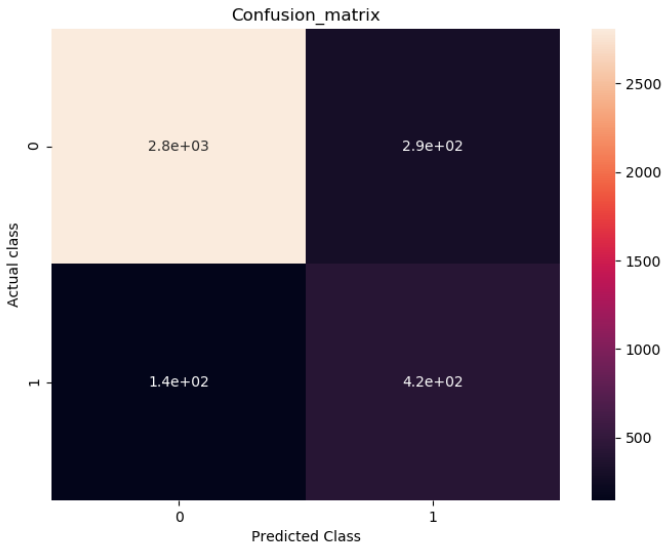
The number 250 ended up being the best parameter to use for our Random Forest model. Since the number was not at either the lower band or higher band of our range, it was not necessary to change our interval and re-run the Grid Search algorithm.

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

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	425
TN	2811
FP	294
FN	144

Performance using a subset of features:

We tested different models adding features based on average percentage of variation explained by each of them (please see list provided on the previous page). We added the features from the highest percentage to the lowest.

number of features	1	2	3	4	5	6	7	8	9	10	11	12	13
True Positives	345	413	416	395	407	420	419	417	417	425	416	426	425
False Positives	251	510	437	376	319	333	311	316	307	292	312	301	292
*Percentage of correct prediction	0.58	0.45	0.49	0.51	0.56	0.56	0.57	0.57	0.58	0.59	0.57	0.59	0.59

*Percentage of correct prediction = True Positives / (True Positives + False Positives)

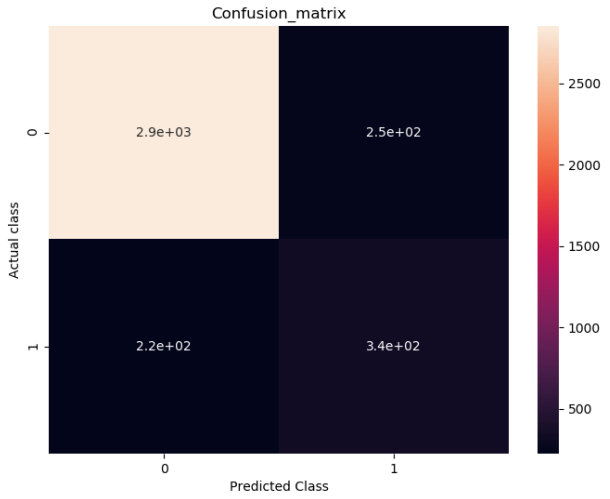
Based on the above figures, it appears that the best model we can provide will only use one feature. Adding more features decreases the percentage of correct prediction and increases the absolute value of False Positives.

By going up to 10 features, for the first time, we can see a negligible gain of 1% in terms of percentage of correct prediction, however this makes our model much more complex for no real added value.

5 – Model building and testing

We can conclude that our best Random Forest model uses only one variable ("Page value") with a Random Forest "n_estimators" parameter of 250.

The performance of the model on the test set is not good, since it can only correctly predict a transaction taking place 58% of the time. The other 42% of the time, it incorrectly predicts a transaction taking place.



TP	345
TN	2854
FP	251
FN	224

6 – Generating Recommendations

Based on the predictive power of the feature used in the final model, we would not recommend the company to deploy this model. The model is not sufficiently reliable to be deployed in production.

However, we can provide the company recommendations based on the analysis made.

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 these pages;
2- Build sales funnel analysis in order to identify the events on the page that caused the customer to leave without making a transaction;
3- Utilising Web Analytics tools to identify the most relevant marketing campaigns that the company could set up to increase its sales

Part 2 (PCA):

1 – PCA Implementation

Through the random forest model, we found that only one feature could be used to build our best model.

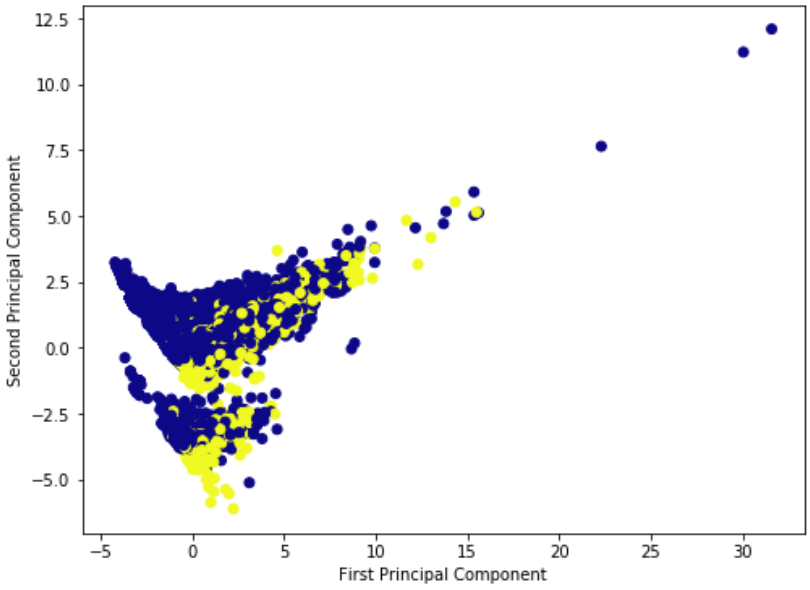
Since PCA cannot be run with only one feature, we don't have a subset of several most significant features. For this reason, we have made the decision to run PCA using all features available in the dataset.

The amount of variance explained by the two first components is:

Component	Variance explained
1st principal component	0.1514 (15%)
2nd principal component	0.1031 (10%)
Sum of 1st and 2nd components	0.2546 (25%)

The two first principal components are only able to capture 25% of the variance of the data. In order to have a reliable and good representation through PCA, the variance explained by the first two components should be at least 70%.

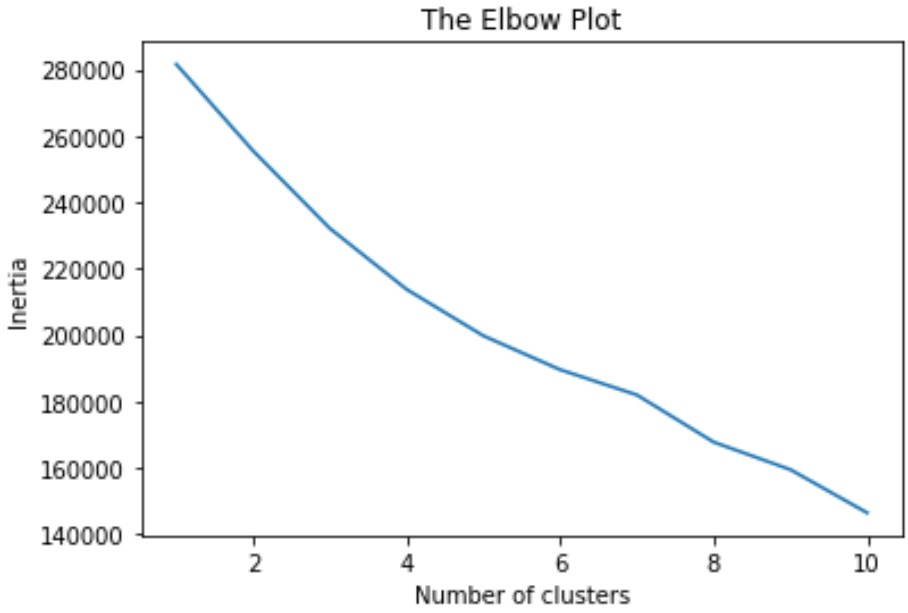
For this reason, we cannot rely on the representation obtained by the PCA on this dataset. It is not a good representation of the data.



As we can see on the plot there is no clear pattern to help us distinguish the two components.

2- Elbow Plot Creation

To guess the number of clusters (different types of categories) in a dataset we analysed the Elbow plot

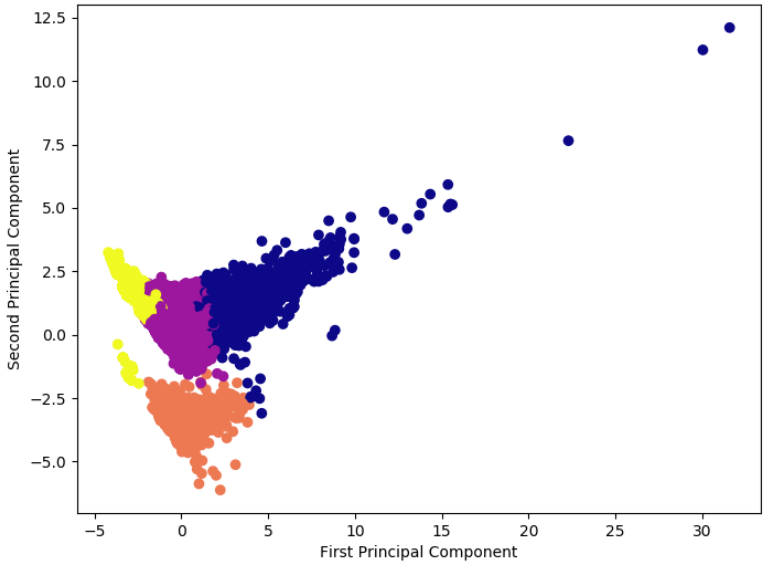


and the values of inertia assigned to each cluster:

Values of Inertia for each cluster:
1 : 281635.0
2 : 255538.87335920575
3 : 232151.18718042172
4 : 213770.48749032422
5 : 199773.787366074
6 : 189548.99555490573
7 : 181940.41436597312
8 : 167720.78159050495
9 : 159424.41492879452
10 : 146396.53395526647

The Elbow plot tells us that we may have around **4 clusters** on the dataset because that is the point after which the inertia starts decreasing in a linear trend. In this case it should make sense trying k-means implementation with k values around 4.

Below an example on how looks like the k-means implementation with 4 clusters:



The k-means is able to build the 4 clusters even if in this dataset they don't have any meaning.

3 -K-Means Implementation

From the dataset we already know that there are only two clusters: one belonging to transaction happening and one belonging to transaction not happening.

Neither the Elbow plot nor the PCA visualization has helped us to build a meaningful k-means plot.