7 Data Analysis

7.1 Introduction

The main goal of the analysis part has been discovering insights about what the customer did, behaved and liked. Those are key aspects to know what to change in our service and how to improve, from the customer journey to the App. To do that we decided to use two different dataset, in order to get different kinds of insights.

7.2 Clustering

The first insight we wanted to address was "What customers did during their stay? What did they liked the most? And what can we propose them the next time they will come?". In order to do that we used a Clustering analysis and a particular dataset from the UCI repository. The dataset is a (249 x 7) .csv file describing the number of reviews each user did. Every row has a specific user ID (249 users), every column reflects a different field.

- **Sports**: Number of reviews on stadiums, sports complex, etc.
- **Religious**: Number of reviews on religious institutions.
- **Nature**: Number of reviews on beach, lake, river, etc.
- **Theatre**: Number of reviews on theatres, exhibitions, etc.
- **Shopping**: Number of reviews on malls, shopping places, etc.
- **Picnic**: Number of reviews on parks, picnic spots, etc.

By the end of the first two months of opening, we assume to have available a similar review dataset about our activities.

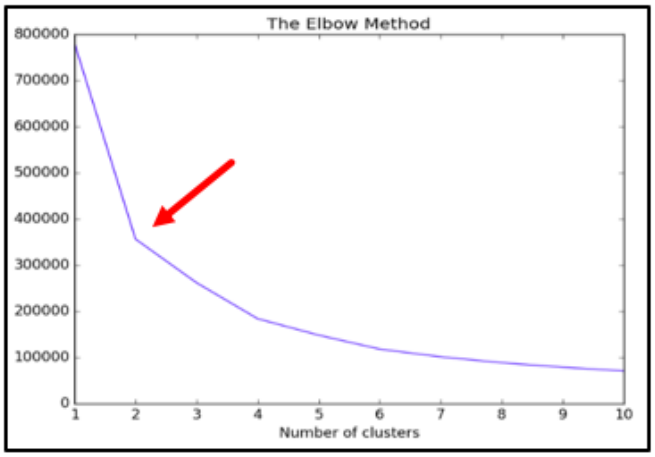
	User Id	Sports	Religious	Nature	Theatre	Shopping	Picnic
0	User 1	2	77	79	69	68	95
1	User 2	2	62	76	76	69	68
2	User 3	2	50	97	87	50	75
3	User 4	2	68	77	95	76	61
4	User 5	2	98	54	59	95	86
5	User 6	3	52	109	93	52	76
6	User 7	3	64	85	82	73	69

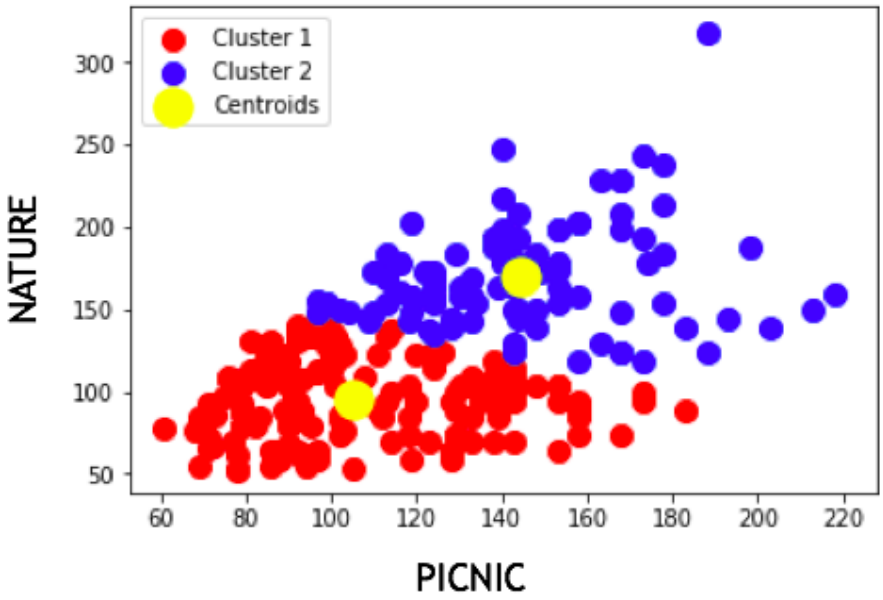
link: https://archive.ics.uci.edu/ml/datasets/BuddyMove+Data+Set

7.2.1 KMeans

In order to perform the clustering analysis we chose to set the number of clusters before the fit. K-means algorithm allowed us to follow this assumption. We applied the Elbow method to choose k. To get info about what activities customers did the most We followed two different paths:

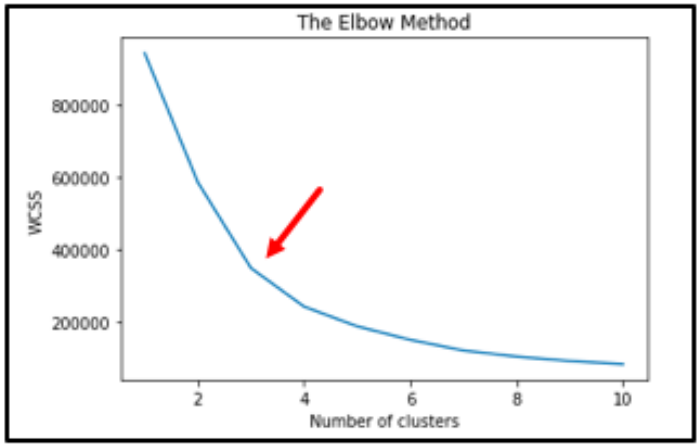
- **Nature - Picnic**:

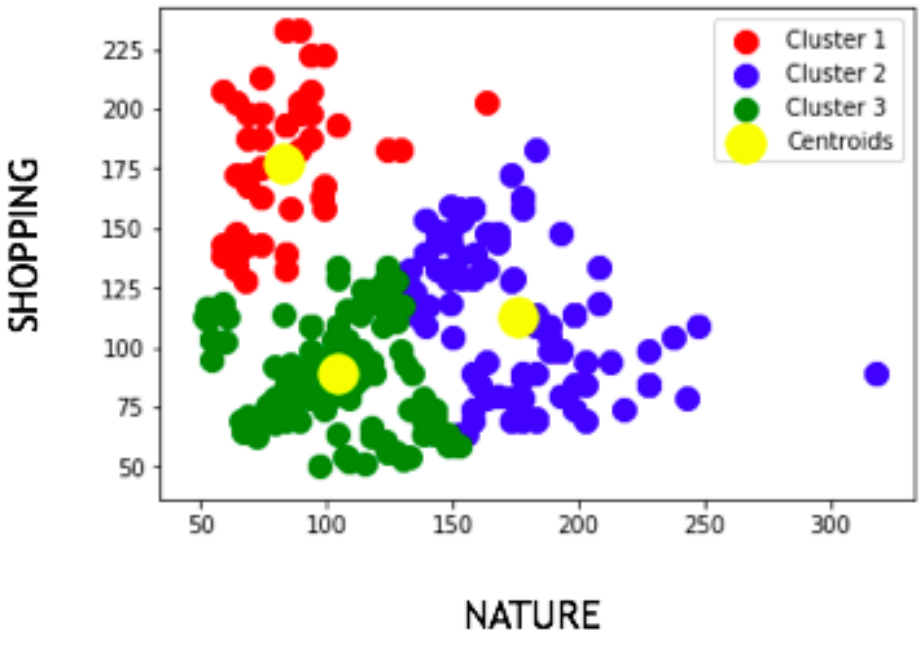




From the plot is clear that people did more picnics than nature activities, as the points are totally growing to the right. Recommending picnics on the app to natural people maybe would be a waste of time.

- **Shopping - Nature**:

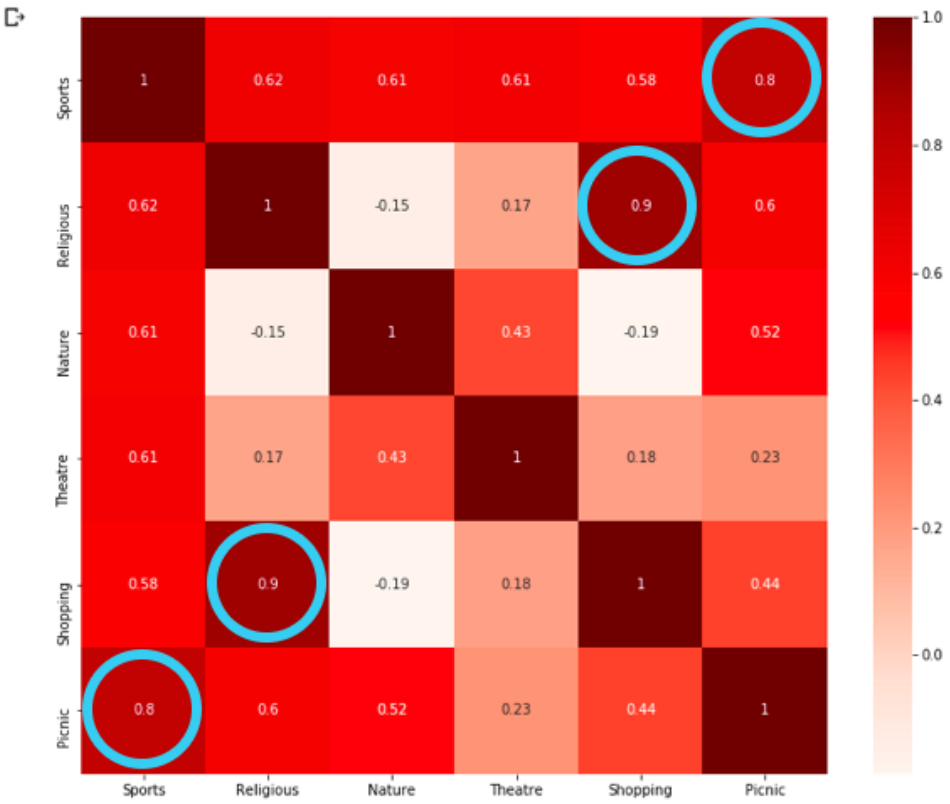




The plot is very different from the previous one: it is reasonable to use three clusters as one of them is clearly apart. It is not easy to understand the correlation between the variables as people seem to behave very differently. Certainly shoppers don't normally go into the wild.

7.2.2 Correlation Matrix

In the end we realized that a clustering analysis was too much to be applied to this kind of dataset and it wasn't even giving us the kind of answers we were looking for. So we opted for a correlation matrix.



This type of plot let us understand if features are actually correlated or not. As seen in the clustering analysis, we have a 0.52 between nature and picnic, while shopping and nature are absolutely not linked at all (-0.19). Remarkable is the fact that also Religion and Shopping seem to be correlated at 0.9 out of 1. That was really unexpected.

7.3 Classification

The second insight we wanted to address was “Are we able to predict if the customer will return to the watershed after his first stay?”. In order to do that we used a Classification analysis and a particular dataset from the UCI repository. The dataset is a (45211 x k) .csv file describing data related to a direct marketing campaigns of a Portuguese banking institution, based on phone calls. The campaign is divided in 3 steps (calls); our goal was to predict the customer behaviour at the end of the first call, without any data from the next calls. The final dataset, indeed, will be a (45211 x 9) .csv file.

- **Age**: age.
- **Job**: type of job.
- **Marital**: marital status.
- **Education**: level of education.
- **Default**: has credit in default?
- **Balance**: balance of the year.
- **Housing**: has housing loan?
- **Loan**: has personal loan?
- **y**: has the client subscribed a term deposit?

After the first stay of a customer, we assume to have available a similar dataset comprehensive of age, job, marital, education, in order to know if that customer will return or not.

	age	job	marital	education	default	balance	housing	loan	y
0	58	management	married	tertiary	no	2143	yes	no	no
1	44	technician	single	secondary	no	29	yes	no	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	no
3	47	blue-collar	married	unknown	no	1506	yes	no	no
4	33	unknown	single	unknown	no	1	no	no	no
5	35	management	married	tertiary	no	231	yes	no	no

link: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

7.3.1 Feature Engineering

Dealing with the second dataset required more data cleaning and engineering. First, we deleted all the next calls features, in order to cut all those information we couldn't use. Second we analyzed each variable one by one and decided how to encode it:

	age	job	marital	education	default	balance	housing	loan	y
0	58	management	married	tertiary	no	2143	yes	no	no
1	44	technician	single	secondary	no	29	yes	no	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	no
3	47	blue-collar	married	unknown	no	1506	yes	no	no
4	33	unknown	single	unknown	no	1	no	no	no
5	35	management	married	tertiary	no	231	yes	no	no

CATEGORICAL BINARY BINARY BINARY BINARY

- **Education**: primary, secondary, tertiary (Label Encoded, since we want to keep track of the order. Tertiary brings higher value and score to the user).
- **Marital**: married, single, divorced (binary modified, since there is no need to introduce other categories).
- **Default**: yes, no (binary).
- **Housing**: yes, no (binary).
- **Loan**: yes, no (binary).
- **y**: yes, no (binary).

```
cleanup_nums = {"marital":     {"married": 1, "single": 0, "divorced":-1},
                "education": {"primary": 1, "secondary": 2, "tertiary": 3},
                "default":     {"yes": 1, "no": 0},
                "housing":     {"yes": 1, "no": 0},
                "loan":     {"yes": 1, "no": 0},
                "y":     {"yes": 1, "no": 0}}
```

```
dataset.replace(cleanup_nums, inplace=True)
```

- **Job**: management, technician, etc (Categorical variable encoded as Dummy).

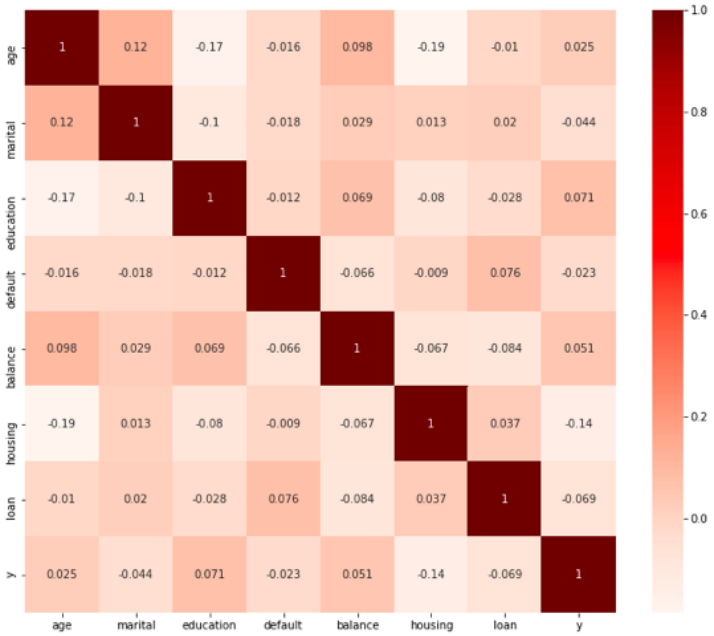
	age	job	marital	education	default	balance	housing	loan	y
0	58	management	1	3	0	2143	1	0	0
1	44	technician	0	2	0	29	1	0	0
2	33	entrepreneur	1	2	0	2	1	1	0
3	47	blue-collar	1	unknown	0	1506	1	0	0
4	33	unknown	0	unknown	0	1	0	0	0

DUMMY

default	balance	housing	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_management
0	2143	1	0	0	0	0	1
0	29	1	0	0	0	0	0

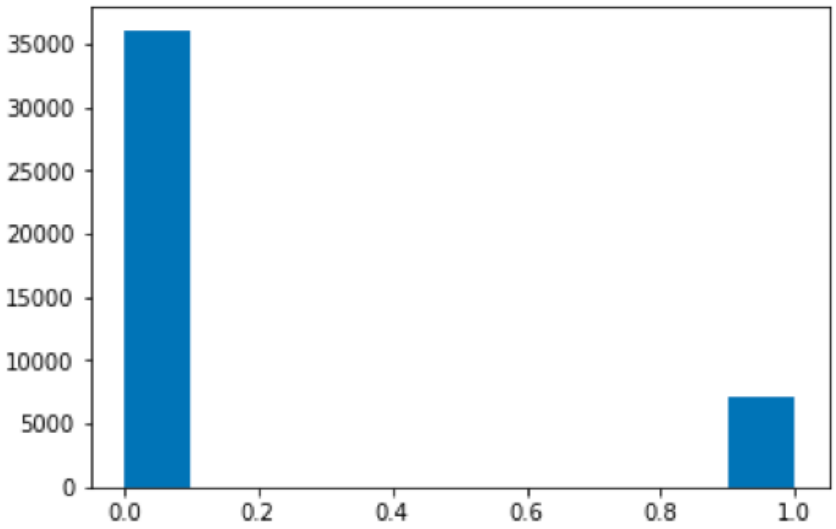
7.3.2 Correlation Matrix

Before any kind of analysis, we tried to understand feature correlation, in order to understand apriori if there were any dependencies. The matrix showed us there were no correlations at all.



7.3.3 Metrics

Since we were dealing with a classification problem, the first common solution we chose for metrics was confusion matrix, so that we could calculate accuracy and precision and use them to rate the models. During the process, however, we realized that we missed an important aspect of our dataset: the target, y, was imbalanced. This event brought us to rethink our metrics and overall every consideration we were going to make. In the end we chose RECALL and F1 as main metrics, with a particular focus on RECALL as we want to identify the completeness of the classifier and lower as much as possible all the false negatives output, in order to focus on the customers who subscribed the service.



7.3.4 Models

The second best way to solve a machine learning problem in general is to try several models and compare their results, as each model is characterized by unique features that, depending on the problem, are able to make it perform better or worse. This is specifically true for imbalanced dataset. In our analysis we focused on trees as decision trees frequently perform well on imbalanced data. They work by learning a hierarchy of if/else questions that normally can be forced to address both classes.

- **Decision Tree Classifier**: DT seemed learning quite well from the dataset and, even if the recall was not so high, averall the F1 score was very curious.

	precision	recall	f1-score	support
NOT-sub	0.87	0.86	0.87	9052
Subscribed	0.31	0.31	0.31	1747

- **Random Forest Classifier**: as a generalization of decision tree, RF should have been performing better. On the contrary, during the process it had a different behavior. We'll discuss more about that after more analysis.

	precision	recall	f1-score	support
NOT-sub	0.86	0.94	0.89	9052
Subscribed	0.36	0.18	0.24	1747

- **K-NN**: Besides others, the third best one was the K-NN. We decided to put it into the documentation to proof the fact we considered also other kind of models, even if the result was worse.

	precision	recall	f1-score	support
NOT-sub	0.85	0.95	0.90	9120
Subscribed	0.29	0.10	0.15	1679

For any doubt, in the notebook each model has also it's confusion matrix, in order to make the reader understand better the overall trend.

7.3.5 Under-Sampling

Third way to address an imbalanced dataset is to re-sample specific features. In our case we lowered the number of rows of the biggest class. Under-sampling, in fact, can be defined as removing some observations from the majority class (the customer doesn't subscribe to the service or doesn't return to the watershed). The selection was performed completely randomly. The only drawback was that we were removing information that may be valuable. For the sake of simplicity (and not to bore the reader with hundreds on analysis) we will focus only on trees.

- **Decision Tree Classifier**: like in the case before, the highest subscribed recall score has been performed by a decision tree. Remarkable is the fact that the value almost doubled (0.65).

	precision	recall	f1-score	support
NOT-sub	0.93	0.58	0.72	9577
Subscribed	0.17	0.65	0.26	1222

- **Random Forest Classifier**: the greatest improvement, however, was performed by RF as from a very low 0.18 it reached 0.61 (lower than DT, but still very similar).

	precision	recall	f1-score	support
NOT-sub	0.93	0.65	0.76	9577
Subscribed	0.18	0.61	0.28	1222

F1 scores were slightly different from the normal case, but they were still keeping the small difference they had before. Between the two models, RF F1 did benefit the most from under-sampling.

7.3.6 SMOTE

In order to reach a complete overview of our analysis, we decided to apply another re-sampling technique, opposite from the first one, based on the concept of over sampling the minority dataset. Synthetic Minority Over-sampling Technique (SMOTE), in fact, is able to generate synthetic (fake) observations based on the real ones from the minor class, in order to reach a high similar volume of samples in both classes.

- **Decision Tree Classifier**:

	precision	recall	f1-score	support
NOT-sub	0.92	0.88	0.90	9577
Subscribed	0.28	0.37	0.32	1222

- **Random Forest Classifier**:

	precision	recall	f1-score	support
NOT-sub	0.91	0.92	0.91	9577
Subscribed	0.33	0.32	0.32	1222

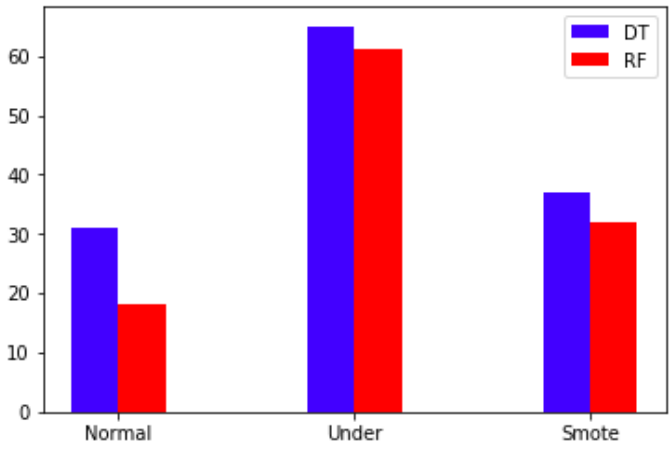
After the algorithm implementation, both models seemed to behave in a similar way. Precision and F1, in fact, were almost the same. RF, however, was the one which did benefit the most.

7.4 Data Analysis Conclusion

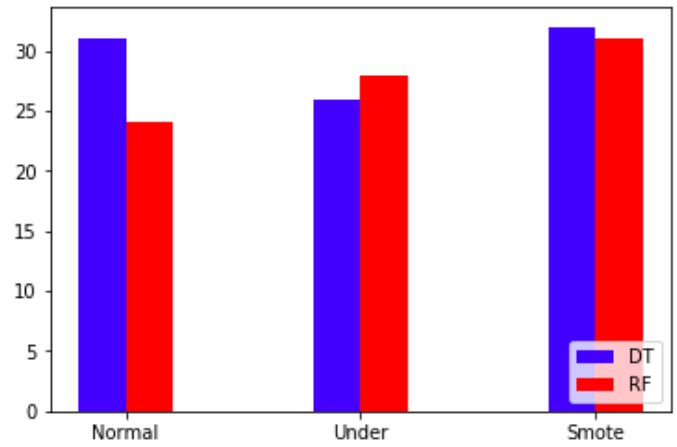
From a brute-force clustering attack we understood that a simple correlation matrix would have been more than fine to get the info we needed out of the dataset. It allowed us to understand connections we knew about (picnic - nature) and the ones we did not (shopping - religion).

The most challenging one has been the last analysis as we had to deal with an imbalanced dataset and decided to follow several paths. At first we studied the different metrics we could use to rate our results and recall was preferred. Then we chose different models to use (decision tree and random forest were the best ones). At last we tried to re-sample our data and perform again each analysis.

- **Recall**



- **F1 score**:



From the beginning we decided to use RECALL as main metric. It is clear that Decision Tree is overall the best model to address this dataset, that combined with the Under-sampling technique provided the best results.

However, considering the F1 score, it is interesting to notice how random forest model improved across the analysis and almost reached the same value as the decision tree, which, on the other hand, seems to behave totally fine even from the beginning, without any kind of sampling. This is maybe the proof that DT is effectively our best choice.