DATA SCIENCE INDUSTRY PROJECT
MAST90107 2020 SM2

Report

THE UNIVERSITY OF MELBOURNE

Group 14 - OaktreeAU

Asouv Bahl, 1047339
asouvb@student.unimelb.edu.au

Fuyao Zhang, 813023
fuyaoz@student.unimelb.edu.au

Harita Bala B, 985054
hbalamurali@student.unimelb.edu.au

Minh Trung Nguyen, 1016151
trnguyen@student.unimelb.edu.au

Tingqian Wang, 1043988
tingqianw@student.unimelb.edu.au

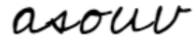
ABSTRACT

Predicting recurrent customers is a key approach in maintaining and enhancing a company's future. In this project, we focused on determining the potential regular donors, mainly, by using different boosting machine learning(ML) algorithms. The labelling of an existing donor as low-value, mid-value or high-value was calculated using Recency, Frequency and Monetary(RFM) features by K-means clustering technique. Tags against each donor in a one-hot encoded format was made used as the feature set after detailed analysis on the data. Real-world problems such as large missing values and imbalanced datasets was handled by choosing the appropriate ML model . In order to classify donors accurately, feature importance was calculated and strategic analysis across different machine learning models was performed.

Before applying different techniques such as above, it is equally crucial if not more, to properly prepare the data to have an appropriate quality. To achieve this strategy, a dashboard was built using R Shiny to provide essential cleaning options for the type of data at hand. The tool with interactive and friendly interface ensures that our client won't need to spend much time on repetitive cleaning tasks and also makes the visualization of data easier with exploratory functionalities.

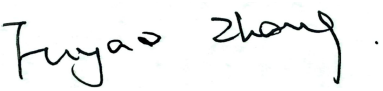
DECLARATION

We certify that this report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of our knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text. The report is 9434 words in length (excluding text in images, tables, bibliographies and appendices).



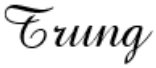
Asouv Bahl

November 1st 2020
Date



Fuyao Zhang

Harita Bala B



Minh Trung Nguyen

Tinqian Wang

ACKNOWLEDGEMENT

In completing our industry project, we had to seek guidance from a lot of people, who deserve our greatest gratitude. The completion of this project gives us immense pleasure. We would like to show our gratitude to **Prof. Michael Kirley** and **Prof. Kate Smith-Miles** , Subject Co-ordinators, University of Melbourne for giving us a good guideline for our project throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in this project.

In addition, a thank you to our project supervisor **Mr. Vivek Katial**, who guided us throughout the year and helped us in overcoming many challenges.

Many people, especially **Mr. Thomas Crawley(Director of Data , Oaktree)** and our team members itself, have made valuable suggestions which gave us an inspiration to improve our work.

Contents

List of Tables

List of Figures

1 Introduction

Oaktree is Australia's largest youth-run international development non-governmental organisation with thousands of donors and supporters. It focuses on funding education and encouraging leadership projects overseas, building the capacity of young people in Australia, and influencing policy change towards youth participation. It has been stated that the organisation has seen a noticeable decline in donations and supporters in the recent years. As the organisation's donations and supporters are the core attributes for the company's successful functioning, it is an important task to develop a turnaround or recovery strategy. The aim of this project is to produce a recovery method to raise both the donations and supporters.

The two main approaches that we focuses in this project are,

1. **Identification of strategies to increase donations to Oaktree:** Obtaining potential and recurrent supporters by analysing characteristics and features of existing donors and supporter base.

2. **Developing a shiny dashboard:** Since our client invests majority of their time in cleaning the data. We provide them with an efficient solution to this problem. The dashboard will not only provide a cleaning functionality but will also be able to provide useful insights. The details of all the functionality that our dashboard provides is elaborated in section (3)

Identification of potential customers with the help of provided supporter databases would be part of a supervised machine learning process where we would focus on labelling the entire list if customers whether if the customer would be eligible as a recurring donors. Data cleaning, feature extraction and labelling would be important factors involved in this strategy. During the implementation of this process in the project in order to avoid repetitive work in the future for the organization, data cleaning scripts will also be designed and provided.

Handling missing data points, developing data cleaning scripts, clustering and labelling potential customers and identifying important features of "high-value" donors are the objectives focused in this project. Solving the mentioned business difficulty with the help of data science techniques is the main focus of the project by providing appropriate insights from the data and presenting in a standard visualization.

2 Literature Review

Classification of potential customers using machine learning techniques has got a significant amount of presence in the recent years. New donor acquisition and current donor promotion are prominent strategies used in classification of fundraisers. In [1], personal attributes such as age, gender, wealth and marital status was used as the features to predict donors acquisition using machine learning models such as Gaussian naıve bayes, random forest, and support vector machine. In the experiment where small donors were excluded from the test set, SVM identified 75.15% of the donors with a precision of 75.92% being the best results that were obtained.

Another interesting aspect of donation promotion has to be identifying larger donators, as one main thing that is observed among many non-profit organisations follow an adhoc work process and so there is no preemptive model. In [2], they built models to predict major gift giving donators and annually subscribed donators. This intends to focus on the idea of identifying high-impact donators as these donations have higher value and importance for the organizations.

Customer segmentation to identify the recurring customers has a lot of different perspectives and approaches to obtain optimal results customised to each use cases. Though customer segmentation fall into a huge category, the techniques used can vary. The technique used in our approach to label the donators were using RFM features for label classification.

The first trait that we faced in classification of recurrent donators was sparse feature set, especially when there is unavailability of customer identification data, which is quite common in a real-world problem. There is a study [3] suggests that the integration of RFM analysis and association rule mining has been proven to be effective while discovering valuable customer purchasing behavior. It is noted that it is difficult to see patterns without the customer identification information. They evaluate the recency, frequency, and monetary scores of frequent patterns to define RFM-patterns. They compose a tree structure to compactly store entire transaction database in main memory and develop a novel method to efficiently discover a complete set of RFM-patterns. It helped them in predicting the purchasing pattern without the customer identification details.

Regarding imbalanced response we can see that in one of the papers , the cardiac surgery dataset is used which is about 4976 cases with died 4.2% and alive 95.8% cases. In this classification problems, the accuracy rate of the model is not an appropriate measure when there is imbalanced problem due to the fact that it will be biased towards the majority class. Thus, the performance of the classifier is measured using sensitivity and precision Oversampling and undersampling are found to work well in improving the classification for the imbalanced dataset. [4]. Several studies have compared the Bagging and Boosting methods. Boosting has been shown to be promising in handling imbalanced data. The case study on predicting customer attrition risk showed that combination of boosting and case sampling can improve logistic regression performance. Undersampling, oversampling, boosting and bagging have all proved to increase the accuracy in imbalanced datasets is observed in various research papers[5]. Though there is no significant proof that boosting is better than sampling techniques.

Another approach used to solve imbalanced response in one of the papers [6] was by stratified sampling which is sampling method that takes into account the existence of disjoint groups

within a population and produces samples where the proportion of these groups is maintained. The stratified version of these two methods is typically used, which splits a dataset so that the proportion of examples of each class in each subset is approximately equal to that in the complete dataset. Labelsets-based stratification is for datasets where the ratio of distinct labelsets over examples is small and iterative stratification is for large ratio. And this paper also uses a new metric for measuring probability calibration under imbalance: the stratified Brier score.

The metrics used to measure accuracy for an imbalanced dataset is also another important aspect while building the model. Looking in to the accuracy metrics and their calculation it is noticed that when precision and recall do not differ much within classes, the difference between evaluating a classifier with one or the other metric is negligible. Since macro F1 is often used with the intention to assign equal weight to frequent and infrequent classes, this paper recommends evaluating classifiers with F1 (the arithmetic mean over individual F1 scores), which is significantly more robust towards the error type distribution[7].

Looking at boosting algorithms used for a classification problem, in [8], it uses Extreme Gradient Boosting (XGBoost) algorithm, to identify atypical patterns and classify 55 participants as healthy subjects or patients with epilepsy. As stated in the paper, XGBoost was chosen due to its significant advantages such as dealing with missing values, requiring data scaling and implying a computationally efficient variant of gradient boosting algorithm[9].

In our project, we have decided to make use of various techniques and approaches from different papers to overcome challenges to obtain the best modelling result to predict recurring donators.

3 Dashboard

Shiny in R was used to develop the dashboard, rather more specifically *shinydashboard* package in R was used. The *shinydashboard* package provides a basic framework for the dashboard. The dashboard that we developed has three modules.

These are:

1. Correlation Module
2. Clustering Module
3. Cleaning Module

The first two features are used to visualize the correlations in the data and to showcase our cluster analysis. The last and also the most important one is the Cleaning Module, which is asked by the client, allows our client to do the cleaning by himself and in a more efficient manner.

3.1 Correlation Module

Understanding the correlation between variables is an essential part of every analysis and we provide an easy way to do so with the help of our correlation module.

The module allows users to select any two variables (or columns) from *Nationbuilder* dataset and outputs a scatter plot. The scatter plot in this module is generated using *ggplot2* library of R. The scatter plots generated are useful in understanding the correlation between any two user-chosen variables. The client can use this as an exploratory tool to discover any correlation between the variables in their dataset which can guide them to carry out further analysis in the future.

Figure (1) shows an image of the correlation module.

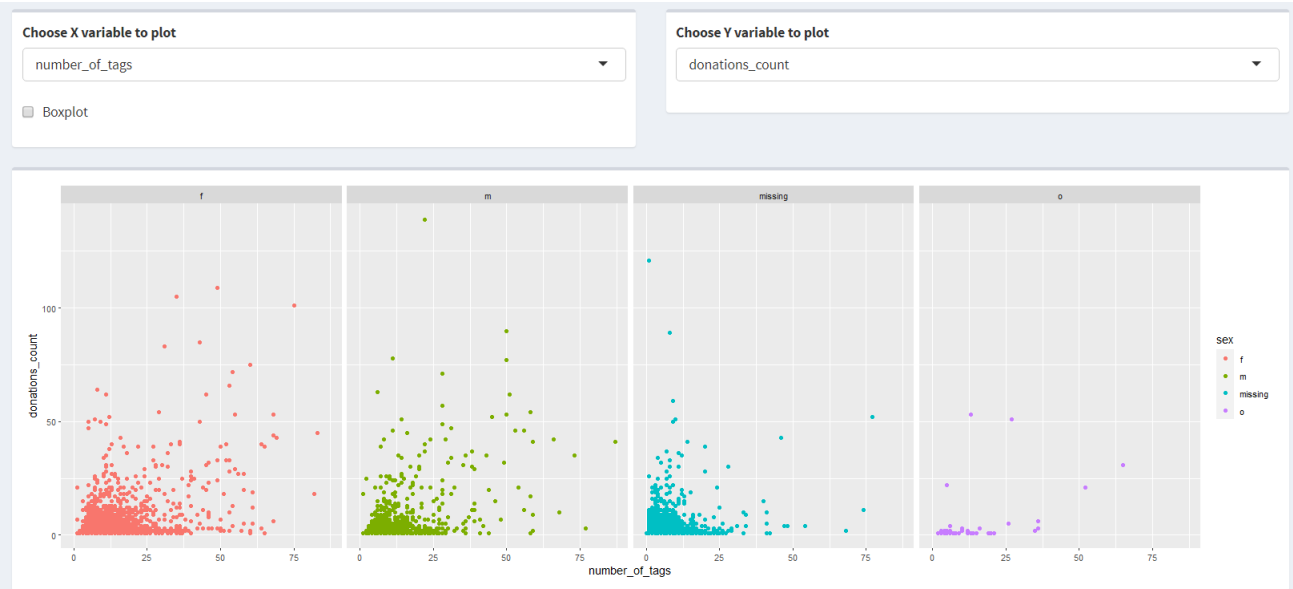


Figure 1: Correlation Module

The module draws scatter plots for the two selected variable input with different facets based on gender. A check box option is provided to draw a boxplot when a categorical variable is plotted against a continuous one.

The figure (2) helps in understanding the functioning of the module.

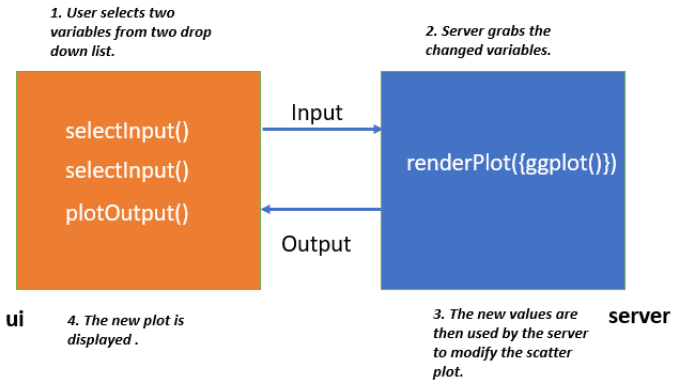


Figure 2: Correlation Module

The plot is interactive in which the user can choose different variables and then the plot will be rendered accordingly. This is done with the use of reactive functionality in R shiny. In this case, the arguments given to the ggplot are reactive values from the drop-down selection boxes using the selectInput() function.

3.2 Clustering Module

The figure(3) is the screenshot for the clustering module in our dashboard.This module allows the user to see various cluster plots. The detail of these graphs are described in the section (4.4.1). The figure(4) describes the working of the module.

Figure 3: Clustering Module - Dashboard

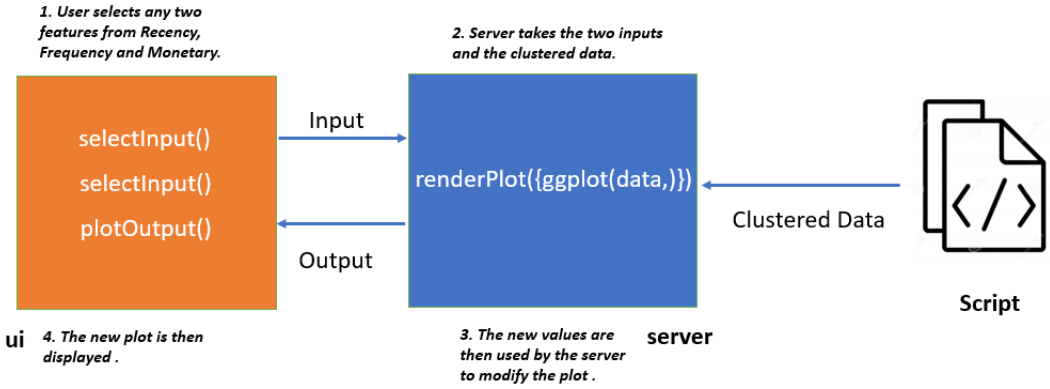


Figure 4: Clustering Module

3.3 Cleaning Module

The Cleaning module is essentially the main part of our dashboard. Currently, our client has to perform the cleaning process repeatedly and manually. He usually does it in Excel which is time-consuming and tedious for some tasks. Therefore, he has asked us to provide a dashboard that he can use to carry out those tasks more efficiently.

The features included in our Cleaning app are based on our client's needs. Indeed, to carry out a useful analysis, we need to have data of good quality. Since Oaktree doesn't have an integrated Data Warehouse, the quality of data is quite bad. Therefore, when someone wants some pieces of the data, our client often has to manually do it in Excel which is slow and inefficient given the size of the data. As a result, we believe that our dashboard will help our client to reduce the workload in cleaning the data and better preparing them for further analysis. In the end, "Rubbish in, rubbish out", no matter how good your analysis methods are, you will not get any insights if your data is inappropriate. Figure (5) is the screenshot of the Cleaning app.

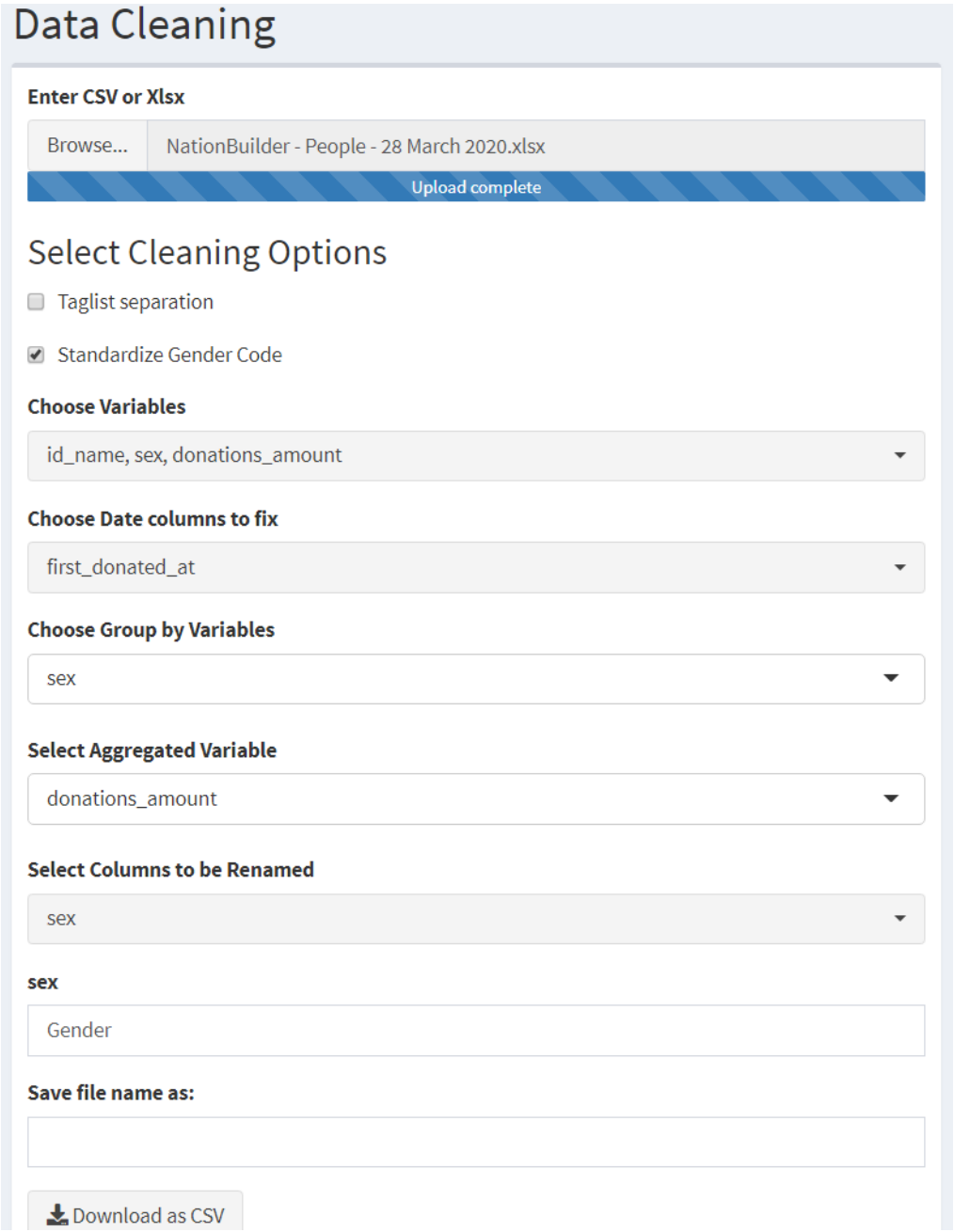


Figure 5: Screenshot of the Cleaning app

The module requires an input file from the user. The input file can only be a *CSV* or an *xlsx* file (as per Oaktree's requirement). To begin, the user needs to upload a data file to the app using the *Browse* button. Next, the module provides two cleaning checkboxes. These are *Tag list Separation* and *Gender Standardization*. The former can be ticked to split the tag list column into one-hot encoded columns. Since there are hundred different tags, this operation takes a while to complete. The latter option can be used to standardize the gender codes to have the same format and cases. After that, several drop-down selection boxes are provided that enable the users to subset the data as well as performing some operations as they want. Firstly, the *Choose Variables* box includes the name of all columns in the uploaded data, which allow users to select columns to display in the output table. The next box gives users the option to choose which *Date* columns to fix. It will transform dates into a single DD/MM/YYYY format. Users can also choose columns to *Group By* and then *Aggregate* on a selected column. Lastly, columns can be renamed to make it easier to understand for some

users(Yet another functionality which was required by our client).

The dashboard then displays the cleaned data with the selected columns in a table format. Figure (6) shows an image of the table output. The table contains several functionalities. Users can sort the data by column by clicking on the column name. They can also filter rows based on the column's values by using the box under each column. If there are more rows than the predefined entries per page, users can switch between pages using the *Next* and *Previous* buttons at the bottom of the table. Finally, when the users are satisfied with the output table, they can download the cleaned data by using the download button and providing a name for the *CSV* file.

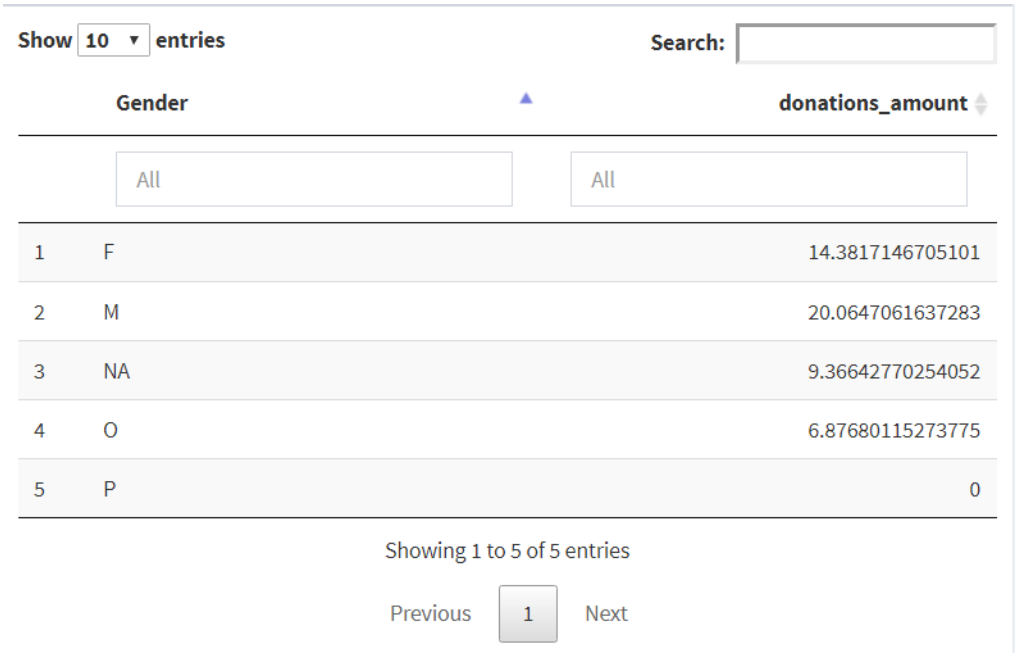


Figure 6: Screenshot of the DataTable output

The figure (7) gives an overview of the functioning of the module.

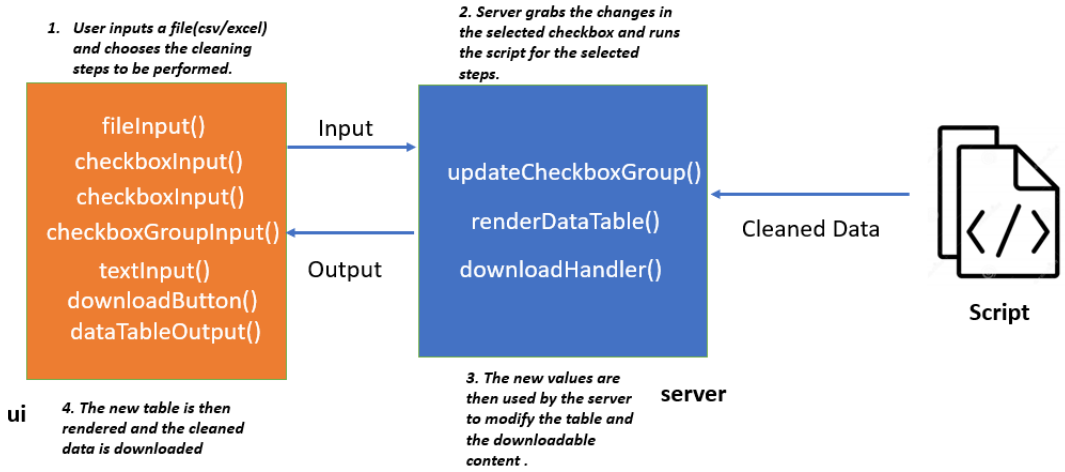


Figure 7: Cleaning Module

4 Unsupervised Learning

We decided to make use of an unsupervised model to discover some groupings of the data. Our objective is to group, within all the people who have donated at least once to Oaktree, donors with similar characteristics together. We hope that we can identify and segment them into something similar to the following clusters:

- **Current Donors**: People who have been donating regularly and frequently to the organization.
- **New Donors**: People donated for the first time recently.
- **Lapsed Donors**: People who have donated before but not recently (during this or the previous year).
- **One-time Donors**: People who donated only once and with a very small amount.

For this, we utilize an unsupervised clustering model called K-means Clustering. Unsupervised learning is used when we do not have the target label to train in our dataset and want to discover the hidden and interesting patterns. K-means is a very simple yet effective model that can quickly segment similar data to the same cluster and we will apply it to our donors' dataset.

4.1 K-means Model

K-means is a classical approach for segmenting a set of data points into K unique and non-overlapping clusters. The mathematical idea behind K-means is that a good set of clusters is when the *within-cluster distance* between each pairwise data points is as small as possible. In other words, points within the same cluster should be as close as possible together. We now define the *within-cluster distance* for cluster C_k as:

$$WSS(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{1}$$

Here, the $WSS(C_k)$ is calculated by summing the squared Euclidean distances between each pair of points in the kth cluster and then divided by the total number of points in it, which is represented by $|C_k|$. And our objective is to assign observations to clusters so that the $WSS(C_k)$ in (1) is smallest.

This is a very difficult problem to solve since there are many different ways to assign n observations to K groups. Luckily, there exists a very simple approach that gives a pretty accurate solution to the K-means optimization problem. The algorithm is shown below:

Algorithm 1: K-means Clustering

1. Randomly initialize K centroids.
2. Using Euclidean distance metric to find the distance between each observation to all of the centroids and then assign each point to the cluster whose centroid is closest.
3. Recalculate the cluster *centroid* for each cluster.
4. Repeat step 2 and 3 until the assignment does not change.

The algorithm above guarantees at least a local optimum solution to equation (1). Figure (8) illustrates the process of the K-means optimization problem by running the algorithm above.

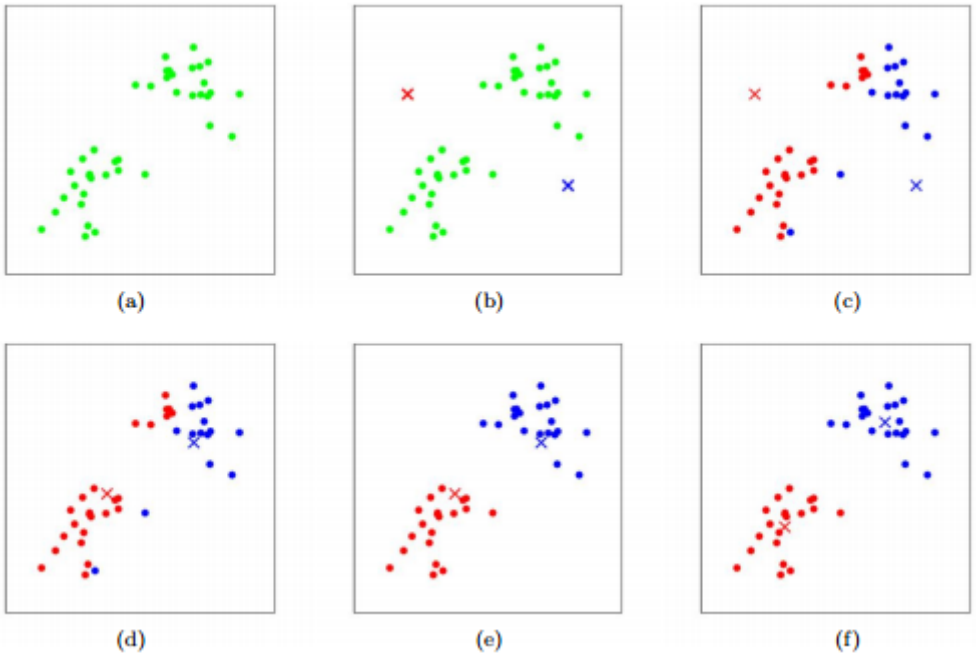


Figure 8: *K-means clustering running with $K = 2$ clusters. Initially (a) has no clusters. The algorithm then randomly choose 2 centroids red and blue cross points in (b). The points are then assigned to the cluster according to the closest centroid using Euclidean distance (c). The process are repeated in (d) until the assignment are stable in (f)*

4.2 Choosing K

In the previous section, we have demonstrated how K-means can partition observations into K distinct groups based on their similarities. But how do we decide the number K? This is one of the disadvantages of the K-means model where you have to choose an appropriate number of clusters K. We mentioned at the beginning of this section that we would like to identify three groups of donors. We can of course follow this hypothesis and choose $K = 3$ to make the model. But there is a simple way which can help us decide which value of K should we choose using again the *within-cluster distance* measure.

Remind that the ultimate objective of the K-means model is to reduce the *within-cluster distance* as much as possible. The idea is that we can run the K-means algorithm multiple times on a range of $K = i, ..., n$. We then choose the value of K which significantly reduces the *total within-cluster distance* summed over all clusters. This jump is often called the "elbow point" in a scree-plot therefore the name "elbow's method".

When applied to our dataset, we found that the elbow happens at $K = 4$ so we decided to use four clusters for our model. The scree-plot of values of K against the *total within-cluster distance* for our dataset is given below:

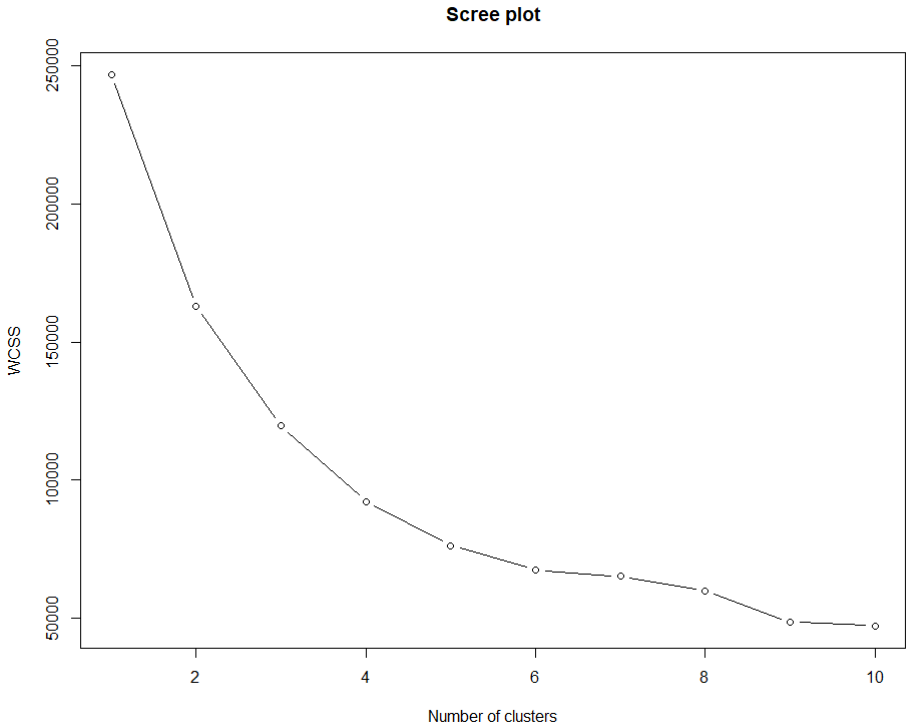


Figure 9: *The elbow seems to happen around K = 3, 4 or 5.*

4.3 Feature Selection

Before building the model, we need to choose the appropriate features for our problem. For this particular customer segmentation problem, we decided to use the Recency, Frequency, Monetary (RFM) features to build our model:

- **Recency**: measures the last time that a person donated to Oaktree. A low score means a recent donor and so we can try to improve their retention.
- **Frequency**: measures the number of times that a person donated to Oaktree. High value implies a frequent donor and can be considered as a loyal customer.
- **Monetary**: measures the amount of money that a person donated to Oaktree. The high value represents a top-donor.

For Recency, we derived from the dataset by subtracting the `last_donated_at` column from the current date. For Frequency, we simply use the `donations_count`. Similarly, `donations_amount` is used as the Monetary feature. We then can build a K-means clustering model based on these three characteristics to partition donors into groups which can then be used as labels for the further supervised classification model. In particular, the centroids mentioned in the algorithm are computed from these 3 features.

The next section shows how we apply the clustering model using two slightly different approaches. One identifies clusters using all three RFM features together. The second method finds clusters separately for each feature which are then combined to form a single overall cluster label. The cluster label is then treated as the value score of a donor. For example, donors assigned to cluster 4 are high-value while cluster 1 means low-value.

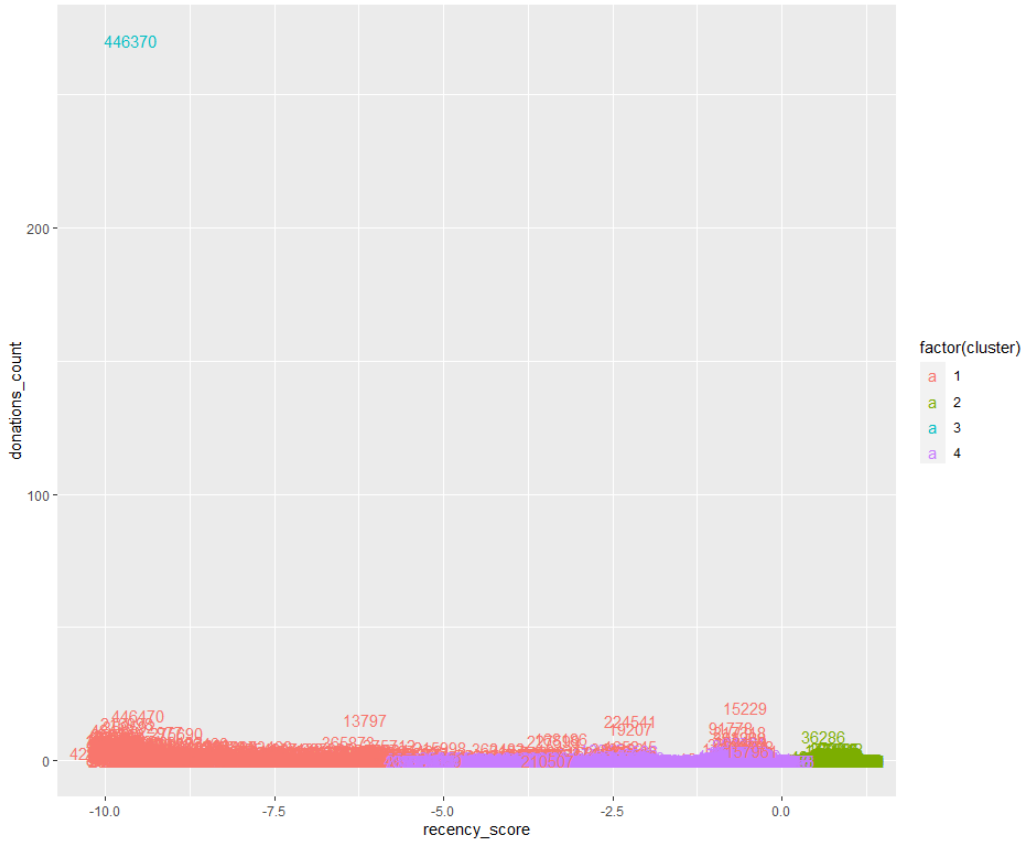
4.4 Experiments & Results

In this section, we present our experiments to partition the data. For the first model, the data is clustered using all three RFM features together. For the second experiment, we attempt to find clusters separately for each R, F and M feature and then manually combine them together later to form a single overall cluster.

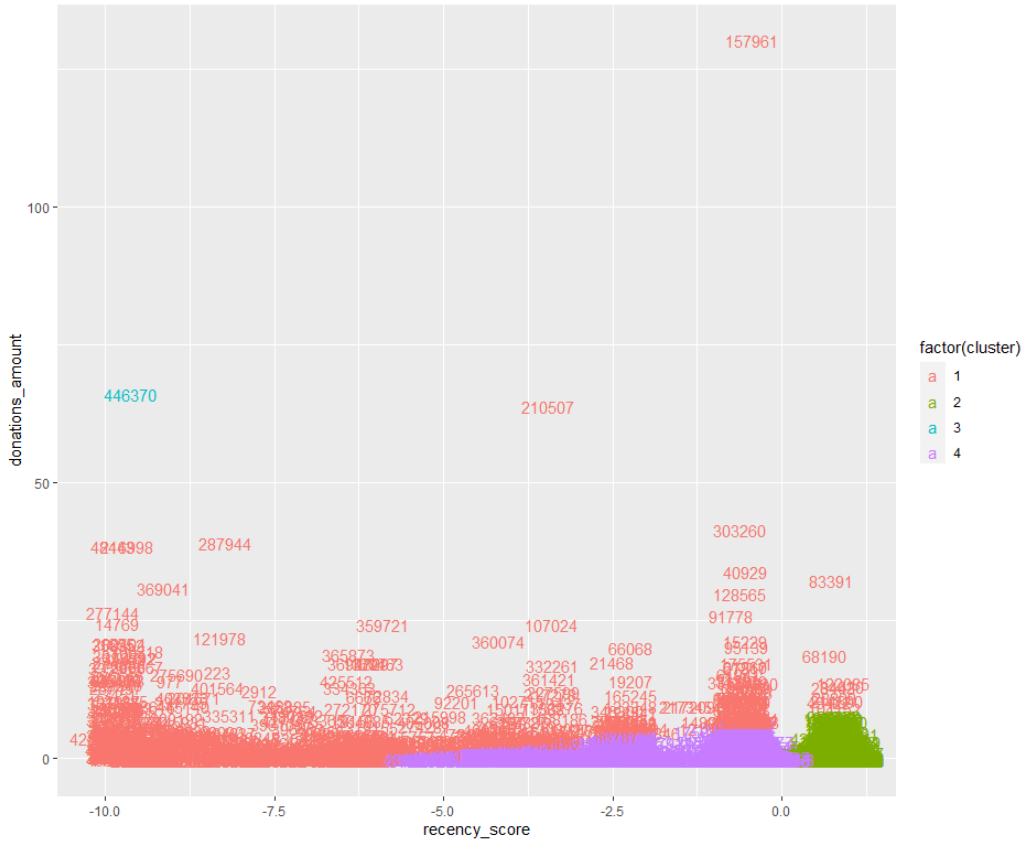
4.4.1 RFM Cluster

As mentioned before, we would need to determine the number of clusters for our model. This can be done by fitting the data to K-means for a range of values of K and choose a value of K which significantly reduces the *total within-cluster sum of square* (WCSS). This happens at the elbow point in a scree plot which plots the number of clusters against WCSS. The plot is shown in Figure (9).

We choose to perform the K-means with $K = 4$. There is just one more step before we can apply the K-means. Since the K-means is very scale variant, all the variables should be in the same scale to produce a reliable result. In our dataset, the range of `donations_count` is very different from `donations_amount`. Therefore, we would need to standardize the variables by centering the data and divided by the standard deviations. The resulting clusters on the standardized data are then shown in Figure (10). The number in the graph is the ID of the donor. Notice that there are two points in the figure which are very far from the others. Specifically, in subplot (a), a person with ID = 446370 has a very high number of donations while in subplot (b), a person with ID = 157961 has a huge amount of donations. In fact, we can already say that these two donors are very valuable for the organization. But since the K-means algorithm is very sensitive to outliers, we should set these two data points aside as special donors and remove them from our model.



(a) Recency vs Frequency cluster



(b) Recency vs Monetary cluster

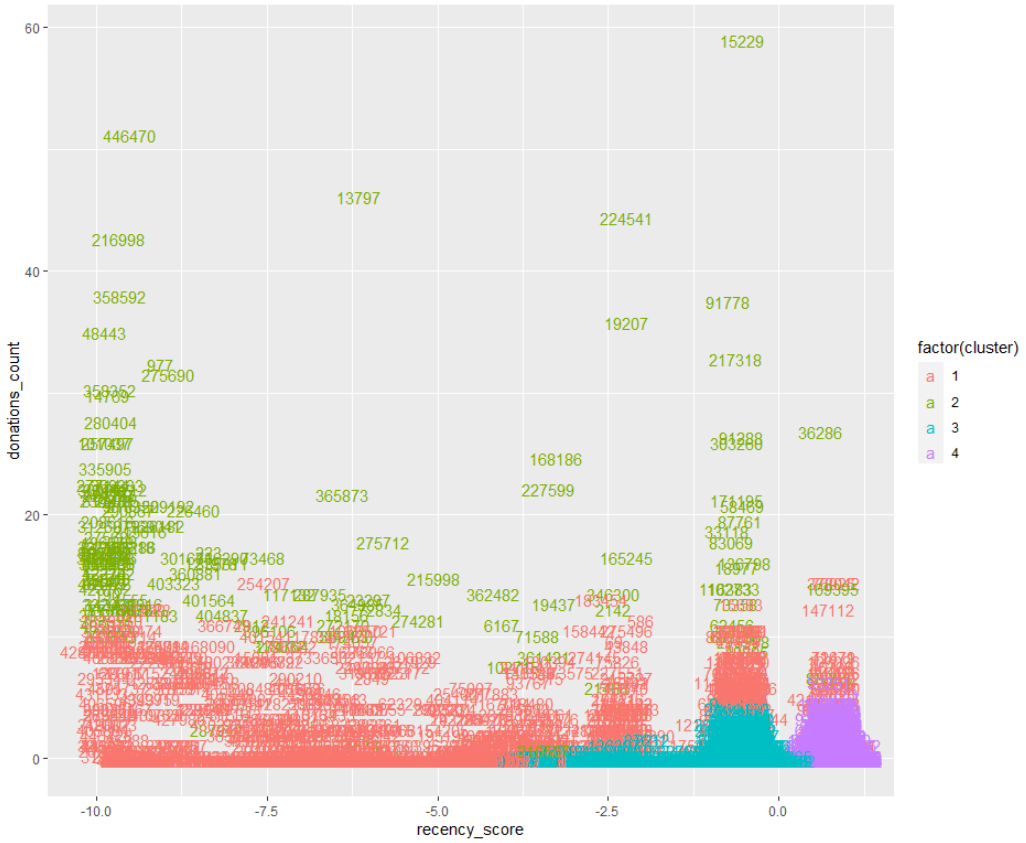
Figure 10: *Clusters plotted with Recency vs Frequency in (a) and vs Monetary in (b)*

The result after removing the outliers is shown in Figure (11). Interpreting the graphs, we can see that people in the green clusters have low recency score and/or high frequency of donations and the amount of their donations. Therefore, these people can be considered as high-value donors and fall into the *Current Donors* category who have been donated recently and frequently with a substantial amount. Thus, Oaktree should focus on this group of donors and try to generate a good relationship with them in order to improve their retention.

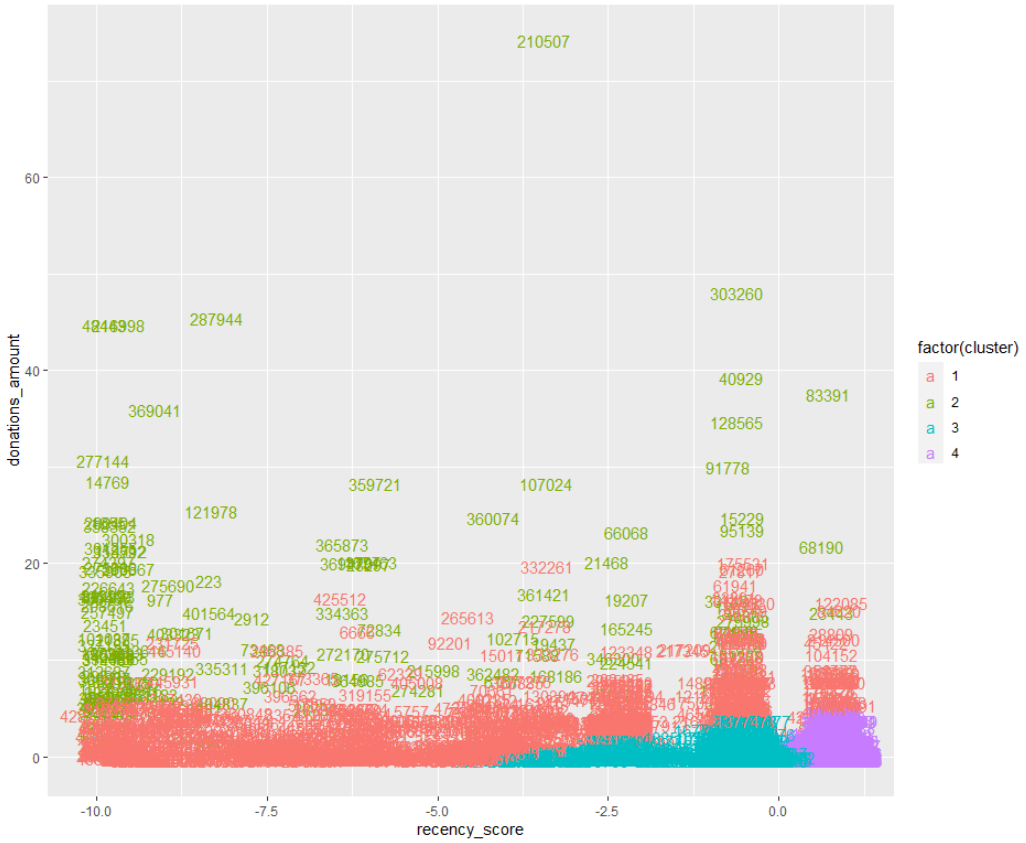
For the blue and purple clusters positioned at the bottom right of the two plots, these people can be considered as One-time donors. It has been a long time since the last time they donated (high recency score) with low donation amount and frequency. These are very low-value donors.

Finally, looking at the red cluster, these people having low to moderate donation amount and frequency which ranges all value of recency. Actually, we want to separate this into two groups, one having a low recency score and the other with a high recency score. Therefore, we try to increase the number of clusters to $K = 5$ to see if it can better partition this group. The result is shown in Figure (12). In this figure, by using five clusters, the model has partitioned the previous red cluster in Figure (11) into two groups. The green cluster at the lower left now consists of people with low recency and low to moderate amount and frequency. These people fall into the *New Donors* group and thus Oaktree can try to slowly approach to encourage them to come back and donate more. The yellow cluster includes people who had donated quite a long time ago with a not so low amount and frequency. Thus, they could be the *Lapsed Donors* that the company can contact to bring them back.

The cluster model presented in Figure (12) is our final model to segment the donors into the four groups that we have proposed at the beginning of this section.

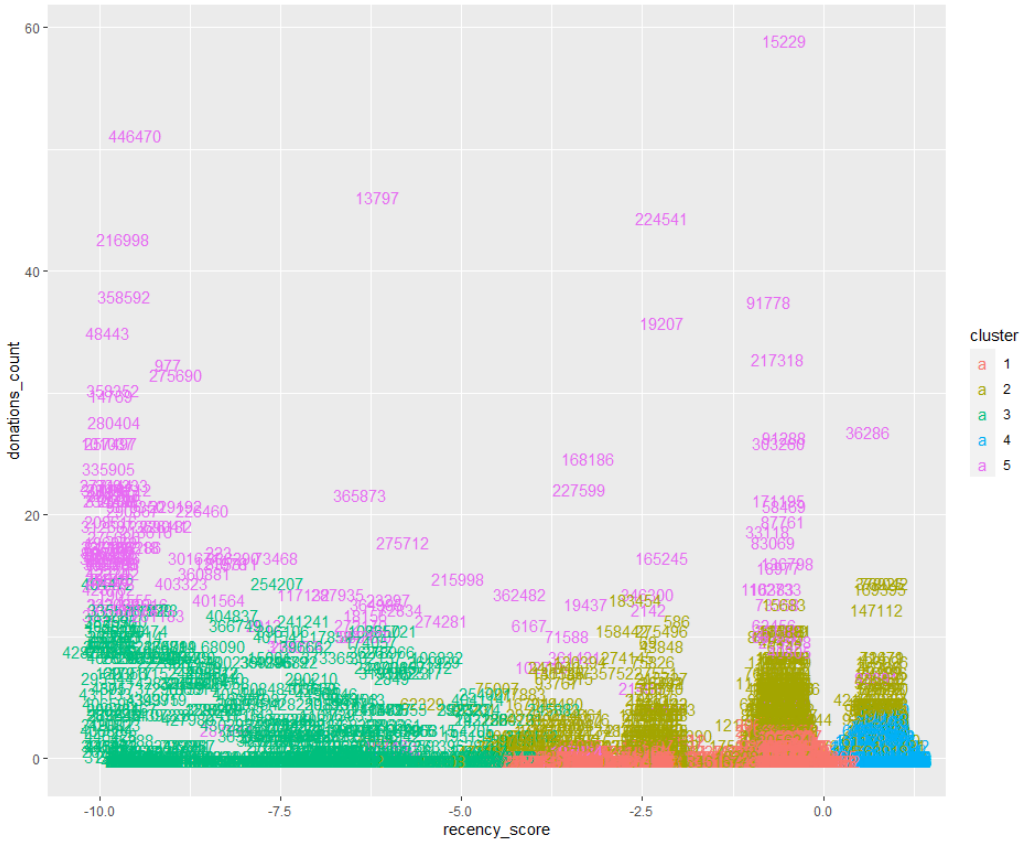


(a) Recency vs Frequency cluster

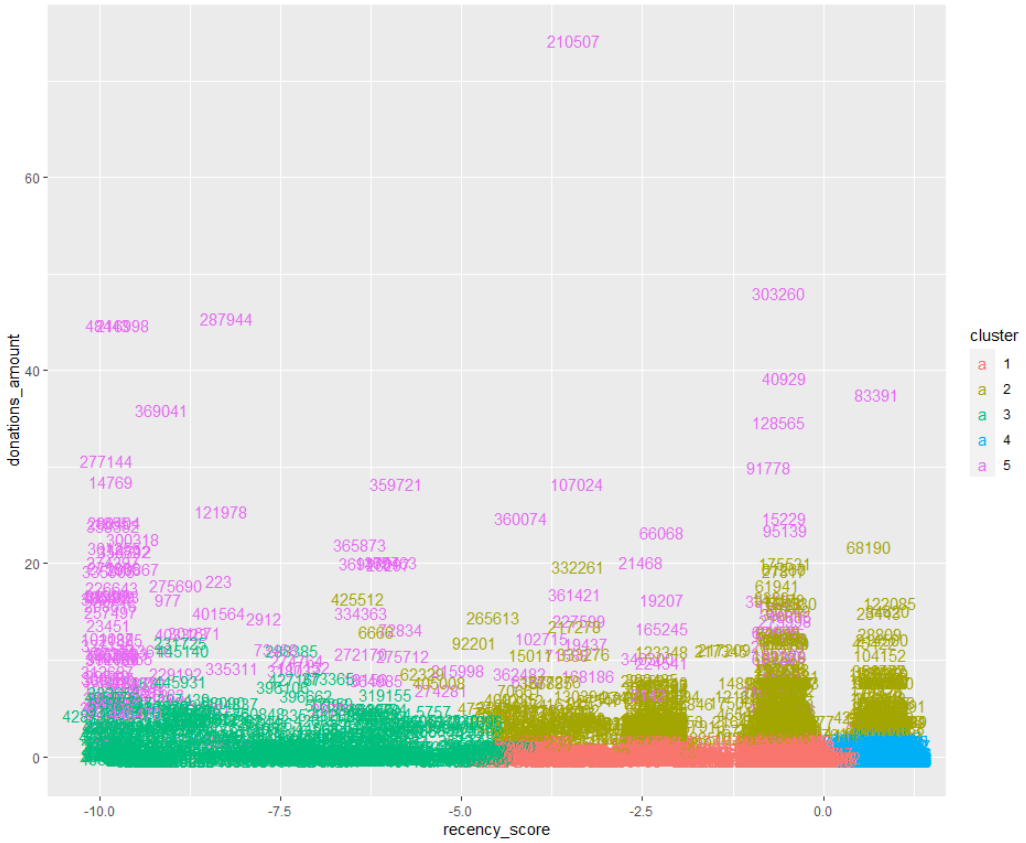


(b) Recency vs Monetary cluster

Figure 11: *Clusters (outlier removed) plotted with Recency vs Frequency in (a) and vs Monetary in (b)*



(a) Recency vs Frequency cluster



(b) Recency vs Monetary cluster

Figure 12: *5 Clusters (outlier removed) plotted with Recency vs Frequency in (a) and vs Monetary in (b)*

4.4.2 Clusters as label

To create the target variable for our supervised model, we again utilise the K-means model but only with a slightly different approach. Instead of clustering based on all three RFM features altogether, we make a separate model for each feature. The cluster labels output will be re-ranked based on the value of the centroids. Consequently, the ranked cluster labels can be seen as the scores which will be used to calculate the overall score for each observation. For example, using $k = 4$ clusters on Recency, after re-ranked, cluster 0 corresponds to the customer with the highest Recency, and cluster 3 is the lowest Recency (since the lower the Recency the better). On the other hand, cluster 0 of Frequency and Monetary corresponds to the centroid of the lowest value. After having all three models, we compute an overall score by summing the cluster labels of the three RFM features. The result is shown by the table [1].

Overall Score	Recency	Frequency	Monetary
0	2912.058875	1.148998	36.126686
1	2554.897035	1.260917	43.729254
2	2390.576923	3.128022	238.374835
3	1821.229358	4.770642	360.565331
4	1350.271739	11.358696	568.339348
5	751.791367	20.877698	971.067266
6	494.416667	40.541667	1045.906458
7	226.000000	51.187500	2304.672917
9	101.000000	1882.000000	9406.000000

Table 1: Overall Score using RFM features

Finally, we assign people with an overall score greater than 6 to the "High-value" group, those greater than 3 to the "Mid-value" group, and the remaining to the "Low-value" group. As a result, our supervised model will be a multi-classification model on 3 classes.

5 Supervised

5.1 Features

The first objective of our project is to identify the features of high value donors and for this particular task we made use of the taglist column in the nationbuilder dataset. The taglist column contains tags for each donor. The one-hot encoded tags separated from the taglist column in the "Nationbuilder" dataset are used as features and the training examples are labelled using the labelling methodology discussed in the section 4.4.2.

There are several reasons why we only use the tag list as the feature to train our model. First of all, the demographic features such as occupation, address, postcode, etc are pretty much missing. Instead, the organization collects some information about the customer through the tags. Therefore, most of the useful information about customers are in the tag list column. Moreover, our client also explicitly asks us to find out which tags are the most important to determine a potential donor. Secondly, we have thought about using RFM as well to train the classification model. But since we have to derive the labels using those features already, it will be inappropriate to use them again to train the model. As a consequence, we take only the one-hot encoded tag columns as the features for our task.

X signifies the design matrix and Y is the response.
X := '2010-2016 Campaigns Deeply Engaged': 1,
'Electorate: Melbourne Ports': 1,
'Event: 10YA 2013: Attended': 1,
'Event: CEO Welcome 2015: Attended': 1,
'Event: CEO Welcome 2015: Invited': 1,
"Event: CEO Welcome 2015: RSVP'd Yes": 1,
'Event: Future Fund Launch 2008: Attended': 1,
'Event: Neima 2015: Guest: Attended': 1,
'Event: Neima 2015: Guest: Invited': 1,
'Has: End Poverty 2014: Roadtrip': 1,
'Has: End Poverty 2014: Roadtrip: Paid': 1,
'Has: LBL 2012: Participant': 1,
'Has: LBL 2013: Fundraiser: 1000+': 1, ...

Y := 'lowValue', 'MidValue', 'HighValue'
Since we have an imbalanced response variable, we stratified the train and test sets. The model was trained on 70 percent of data and then tested on the rest 30 percent.

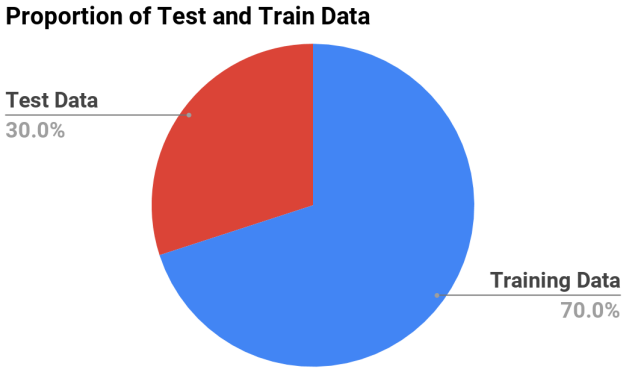


Figure 13: Proportion of Test and Train dataset

5.2 Decision Tree

We use the Decision Tree as our baseline model. It consists of nodes, edges and leaf nodes. Each node is a test of one attribute and the edges are the outcomes of the test. The terminal nodes will give the final result of the classification.

In this method a set of training examples is broken down into smaller and smaller subsets while at the same time an associated decision tree gets incrementally developed. Hence Decision Trees follow **Divide-and-Conquer** Algorithm.

Decision Tree has some advantages which may be suitable for our task. It does not need huge amount of data to train. Additionally, missing values won't affect much on building the tree. It is very intuitive and can be easily explained.

The evaluation metric being used in our task is macro average f1-score, because each classes' precision and recall are all important for us. After 30 iterations, the macro average f1-score is about 0.7183. According to the figure of performance below, we can find that the f1-scores of class "High-value" and class "Mid-value" are very low compared to the one of class "Low-value". One of the reasons is that we have extremely imbalanced training dataset. In our dataset, most people only donate small amount of money such as 20 dollars. Hence most of our data describe the attributes and features of people who is labelled as "Low-value". The second reason that our performance is not very effective is because of the sparsity of the features. Our features are tags separated from the tag list. These tags are the events people attended. However, each person may only attend a few events so in their feature vectors there are a lot of zero categories and these challenges have been discussed in detail in challenges section (6).

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	0.99	24249
Mid-Value	0.65	0.56	0.60	349
High-Value	0.60	0.48	0.53	71
accuracy			0.99	24669
macro avg	0.75	0.68	0.71	24669
weighted avg	0.99	0.99	0.99	24669

Table 2: Decision Tree classification report

Moreover, in the progress of building a tree, it is very easy to be overfitting. We tuned the hyperparameters such as the depth of the tree to reduce overfitting. As stated by [10], more trees will prevent overfitting and provide a stronger model. Hence we used ensemble learning techniques to generate multiple trees to provide a better prediction. We implemented the Decision Tree model using both bagging and boosting techniques separately.

5.3 Random Forest

As we can see from the section [5.2] the decision tree classifiers does not do a good job in classifying the donors. One reason behind this can be that the decision trees are not flexible when it comes to classifying a new sample.

The mentioned disadvantage is overcome by the **Random Forest Classifier**. It is a *bagging* technique. Bagging refers to $Bootstrapping + Aggregation$ to make decisions. The random forest classifier uses both of these techniques and its working can be explained as follows:-

- **Step 1:** Create a bootstrapped dataset. The bootstrap dataset is essentially of the same size as the original dataset and is made by randomly selecting samples from the original dataset(and the same sample can be used more than once.)
- **Step 2:** Create decision tree using the bootstrapped dataset. Subset of the features are picked for each node and the best split is calculated. Since, not at all of the original dataset is covered in the bootstrapped set the remaining samples from the training set(out-of-the-bag) are used as to calculate error of the tree. The Random forest is dependent on the splitting criteria for making decisions. The split criteria that we used is **GINI** split. It gives an intuition about how good or bad a split is and the equation for it is given as follows:-

  $$GINI = 1 - \sum_{i=1}^{n} (p_i)^2$$

  where p_i is the probability of an object being classified to a particular class.
- **Step 3:** Go To step 1 and repeat n number of times.

Hyperparameter tuning The set of parameters chosen for tuning are tabulated in the table [3]

Parameters	Values	Best Value
n_estimators	[100, 200, 300, 1000]	300
min_samples_split	[2,4,8, 10, 12,16]	10
min_samples_leaf	[1,3,5,7]	1
max_depth	[10,40,60,80, 90, 100, 110]	100

Table 3: Random Forest Parameters

The average macro F1-score we get after running the tuned random forest for 30 trial : 0.74596970

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	1.00	24249
Mid-Value	0.72	0.54	0.61	349
High-Value	0.80	0.52	0.63	71
accuracy			0.99	24669
macro avg	0.84	0.68	0.75	24669
weighted avg	0.99	0.99	0.99	24669

Table 4: Random Forest Classification Report

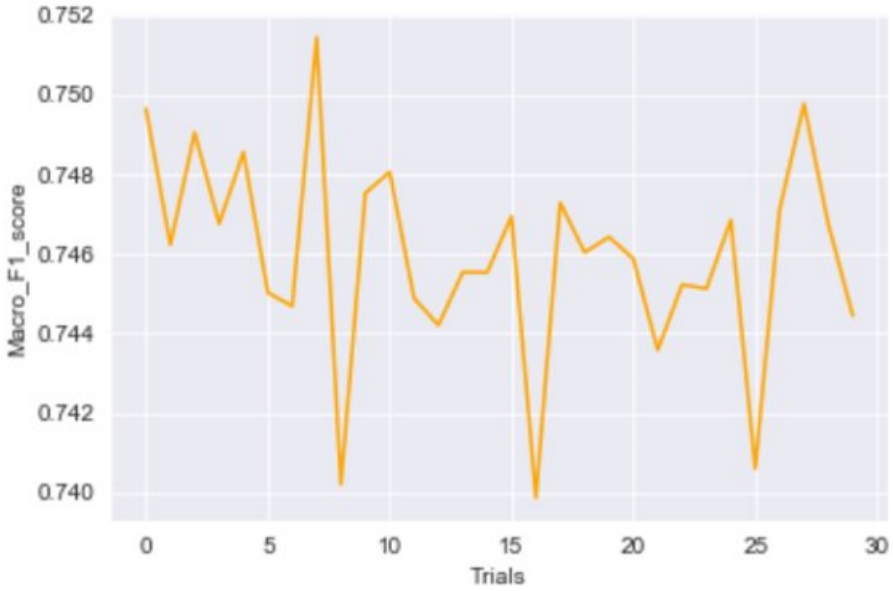


Figure 14: Random Forest for 30 trials

5.4 Gradient Boost

Gradient boosting is an ensemble learning technique which implements the base learners and use them to come out a better prediction. Boosting technique is different from Bagging technique since it trains the base model sequentially not parallelly. It consists of many decision trees and each decision tree is dependent on the previous tree. The model iterates multiple times and the final result will be the weighted sum of each round's results.

The gradient boosting algorithm will identify the shortcomings of the base learners by calculating gradients in the loss function and then improve them in the next round. Firstly, it initializes the predicted value with a constant value using the loss function.

$$F_0(x) = argmin \sum_{i=0}^{n} L(y_i, \gamma)$$

Then in each iteration (m), it calculates the negative gradient of the loss function of the previous tree. The loss function can be the square error loss and then the r_{im} will be two times residual error.

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$$

$$\text{for i} = 1, ...n; m = 1, ..., M$$

Each decision tree fits on the modified version of training data $(x_i, r_{im})_{(i=1)}^{n}$. After fitting the decision tree $h_m(x)$ using the negative gradient in each iteration we will calculate a multiplier γ_m by solving the following optimization problem.

$$\gamma_m = argmin \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

Then use the optimized γ_m and the previous state's model to update the model using Forward Stage-wise Additive Model.

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x)$$

After the M iterations, we get a final model $F_M(x)$. Since we keep optimizing the model by reducing the loss, the final model $F_M(x)$ give better performance. In this process, the weak classifiers such as Decision Tree are combined as a stronger classifier. It can be shown using our example, after 30 iterations it gives 0.735 f1 score on average which is better than the one of baseline model. Particularly, both precision and recall have a slight increase. This improvement is expected since the gradient boosting algorithm can improve the shortcomings of the base learners sequentially.

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	1.00	24249
Mid-Value	0.67	0.57	0.62	349
High-Value	0.71	0.48	0.57	71
accuracy			0.99	24669
macro avg	0.79	0.68	0.73	24669
weighted avg	0.99	0.99	0.99	24669

Table 5: Gradient Boost Classification report

5.5 LightGBM

Since the results from Gradient Boosting Classifier are still not very satisfied for us, we tried two more powerful algorithms which are XGBOOST and LightGBM. We will discuss their structural difference and compare their performance. Both these two algorithms are based on the gradient boosting technique.

LightGBM is a gradient boosting decision tree with two novel techniques applied which are Gradient-based One-Side Sampling and Exclusive Feature Bundling. A gradient boosting decision tree has to scan all the data instances for each feature to calculate the information gain for splitting a tree. This makes the time complexity of gradient boosting model become very high when handling big data. The time complexity is proportional to the number of features and the number of instances. Then GOSS (Gradient-based One-Side Sampling) and EFR (Exclusive Feature Bundling) can improve the efficiency significantly. These two techniques will reduce the number of data instances and features in a reasonable way.

GOSS down samples the instances on the basis of gradients. Data instances with greater gradients should contribute more to the information gain since they are undertrained. GOSS will keep the instances with large gradients and performing random sampling on the ones with small gradients [10]. Since our training dataset has a sparse feature space, EFR will

help to reduce the number of features to speed up the learning and still keep the accuracy. It identifies the features which can be bundled together and then merge them.

These two techniques let LightGBM have high efficiency than other gradient boosting models. It can take lower memory and time to handle even a very large size of data.
Moreover, the way in which the trees are split also makes the LightGBM Classifier superior than many other gradient boosting learners. As shown by the following figure, LightGBM Classifier expand the tree leaf-wise with the best fit and grow the tree vertically while other boosting algorithms expand the tree level-wise and grow it horizontally. It means for every split the algorithm will consider a split's contribution to the global loss instead of only the loss along a particular branch. Hence when growing on the same leaf, the leaf-wise algorithm can reduce more loss and learn faster than the level-wise algorithm.

Figure 15: Level-wise and Leaf-wise growth

From the test result we can see that LightGBM gives a better performance indeed. After 30 iterations, the macro average f1 score is 0.758 approximately. The precision of classifying "High-value" improves a lot which means the ability of correctly classifying data into "High-value" class is enhanced. Both class "High-value" and class "Mid-value" have a slight increase in f1-score. However, our recall scores are still not very good. It means it's hard for the model to learn the features of "High-value" class and "Mid-value" class. The performance is still affected by the poor quality of the training data.

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	1.00	24249
Mid-Value	0.68	0.57	0.62	349
High-Value	0.76	0.58	0.66	71
accuracy			0.99	24669
macro avg	0.81	0.72	0.76	24669
weighted avg	0.99	0.99	0.99	24669

Table 6: LightGBM classification report

From the confusion matrix we can see a lot of "Mid-value" instances are classified as "Low-value". This may be caused by the unclear boundary between "Mid-value" class and "Low-value" class. In our training data, people donate low value amount and the ones donate mid value amount may have similar features.

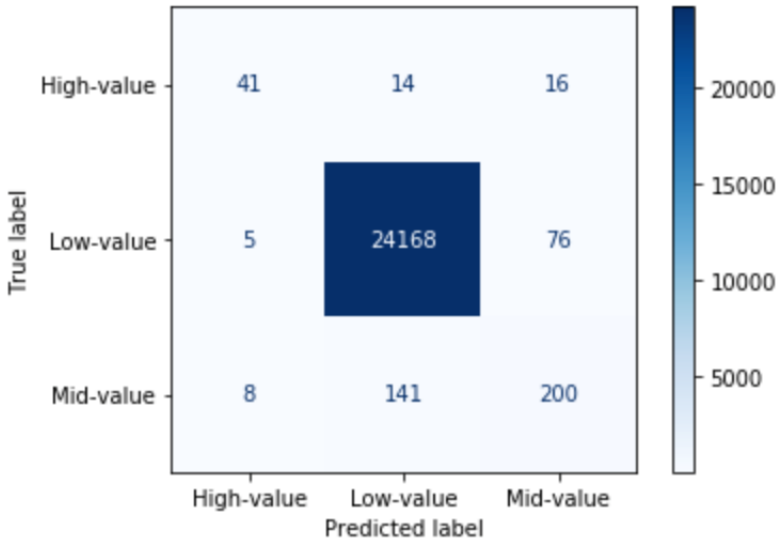


Figure 16: LightGBM Confusion Matrix

5.6 XGBoost

XGBoost stands for eXtreme Gradient Boosting . We incorporated XGBoost algorithm for modelling our problem. XGBoost was chosen because it has the following advantages:-

1. Prevents overfitting by making use of inbuilt L1(Lasso) and L2(ridge) regularisation techniques.
2. Extremely Fast: XGBoost makes use of the multiple cores of the CPU and provides a parallel execution functionality. This is one of the reasons of using XGBoost over GB.
3. Pruning Trees Effectively: XGBoost splits a node to the max_depth specified.

The objective(as the author mentions in [11]) at iteration t is

$$L^{(t)} = \sum_{i=1}^{n} (l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x_i})) + \Omega(f_t) \tag{2}$$

In equation (2) $l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x_i}))$ is the loss function , in which the y_i is the observed value (known from the training data-set) and $\hat{y_i}^{(t-1)}$is the prediction of i^{th} instance at $t-1$ iteration and $f_t(\mathbf{x_i})$ is the output value at the t-iteration. The other part of the equation is the regularisation.

Since the objective in XGBoost is a function of classification trees (for our problem) and is not in the Euclidean domain, the traditional optimisation techniques cannot be applied.

But, with the help of second order Taylor approximation we can transform the function in the Euclidean domain.

The second-order Taylor approximation is given by [11] :

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2$$

So using the above approximation on equation 2. We get [11]:

$$L^{(t)} \approx \sum_{i=1}^{n}[l(y_i, \hat{y_i}^{(t-1)} + g_i f_t(\mathbf{x_i}) + \frac{1}{2} h_i f_t^2(\mathbf{x_i})] + \Omega(f_t) \tag{3}$$

Where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$
Now since the first term in 3 (i.e $l(y_i, \hat{y_i}^{(t-1))}$ will be constant and removing it gives us a function in euclidean domain which can be minimized.
XGBoost builds the next learners using a *"Exact Greedy Algorithm"*. The algorithm (by [11]) is as follows:-

Algorithm 2: Exact Greedy Algorithm

1. Start with a tree with single node which contains all examples.
2. evaluate the loss reduction due to splitting for all features and values that the feature hold, using:
$$gain = loss(parent) - loss(left) - loss(right)$$
3. The Best split will have a positive gain.

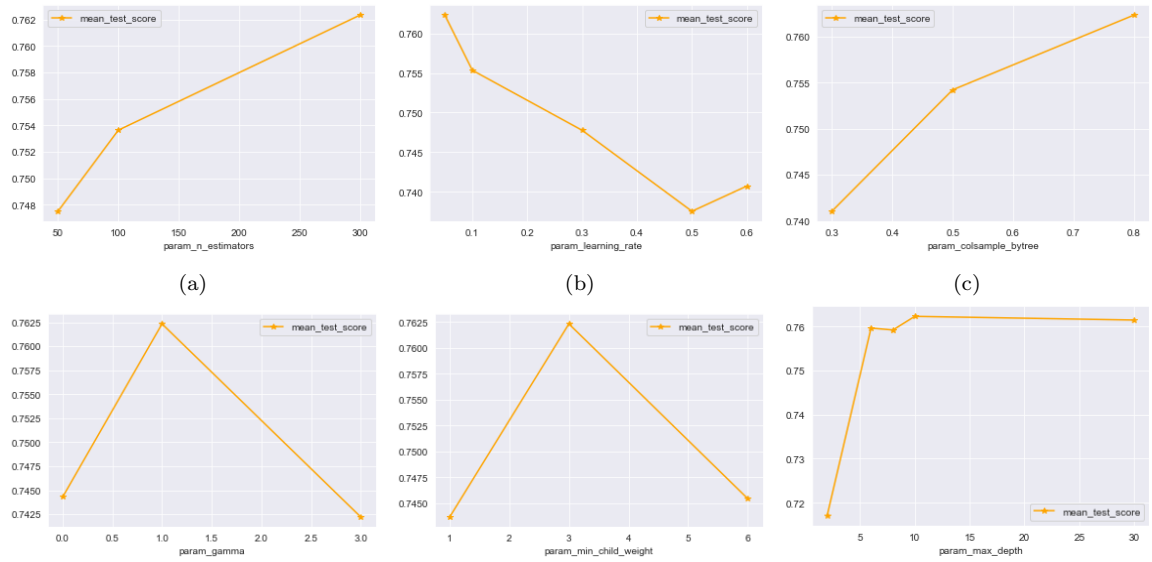
5.6.1 Hyperparameter Tuning

The hyper-parameters that were tuned are as follows: We used grid search approach for tuning the hyperparameters. The **GridSearchCV()** function in python is used for tuning the parameters and the set of parameters chosen to be tuned are as follows(We have tuned on only a small set of parameters because of the time constraints):

Parameters	Values	Best Value
learning_rate	[0.05,0.1,0.3,0.5,0.6]	0.05
max_depth	[2,6,8,10,30]	10
min_child_weight	[1,3,6]	3
colsample_bytree	[0.3,0.5,0.8]	0.8
n_estimators	[50,100,300]	300
gamma	[0,1,3]	1

Table 7: Parameters

The figure [19] depicts how the mean test scores from three cross-validation sets change when the respective parameter changes and other parameters are kept at their best value (that is obtained from GridSearchCV())



(a) (b) (c)

(d) (e) (f)

Figure 17: GridSearchCV() results

The Macro-averaged F1 score we get from XGBoost:-0.77.

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	1.00	24249
Mid-Value	0.69	0.59	0.64	349
High-Value	0.76	0.59	0.67	71
accuracy			0.99	24669
macro avg	0.82	0.73	0.77	24669
weighted avg	0.99	0.99	0.99	24669

Table 8: XGBoost Classification Report

The feature gain importance that we get from XGBoost is as follows:

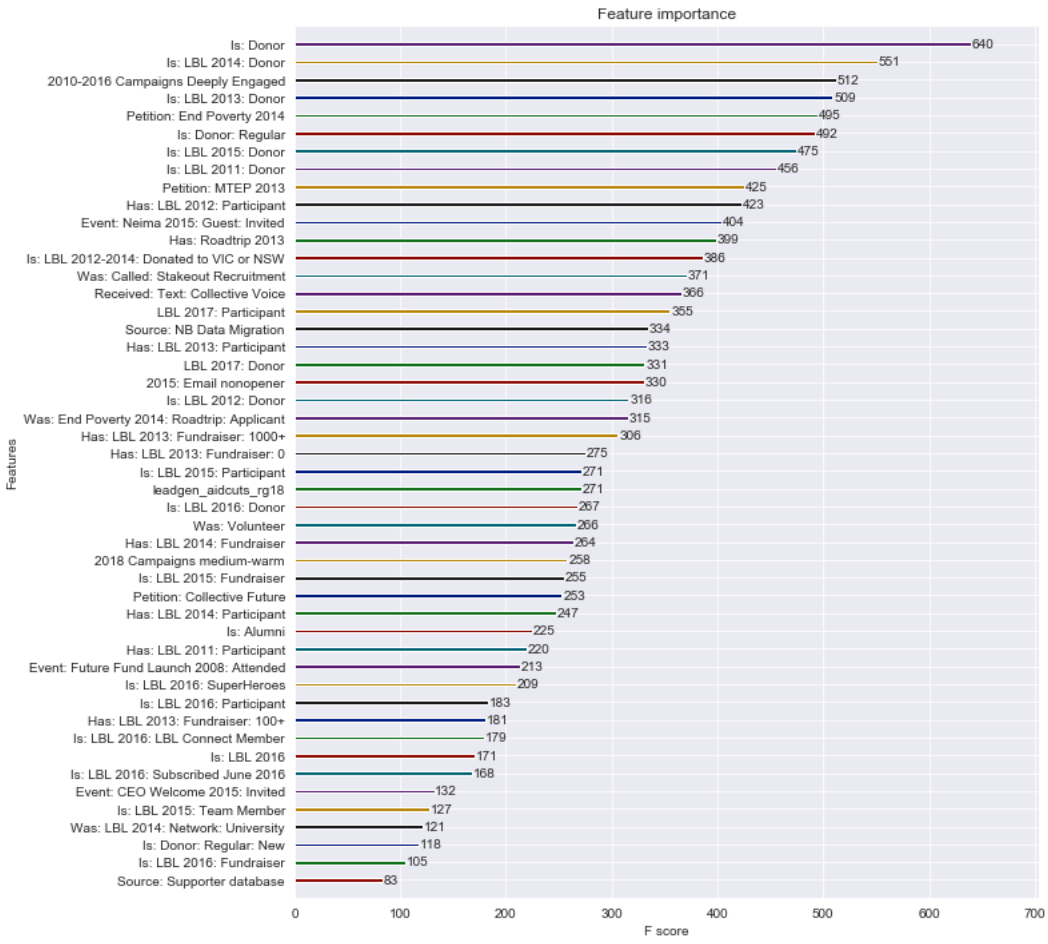


Figure 18: XGBoost Feature Importance

The top 5 features from the figure(18) are as follows:-

- Is Donor
- Is LBL2014:Donor
- 2010-2015 Campaigns Deeply Engaged
- Is LBL 2013 Donor
- Petition: End Poverty 2014

Though the best model we got was using XGBoost, we also tried MLP Classifier.

5.7 MLP Classifier

In this section, we applied ANN to perform the prediction of the value of a donor. Specifically, we applied MLPClassifier on the train and the test dataset, and we also applied permutation importance in finding out the most important features in predicting the value of a donor.

We used the data("rfm_taglist_donorOnly_segmented") which has been labelled with three types of segments generated based on K-Means clustering and RFM analysis. The size of the data is 82,230 of its rows and 89 of its columns.

5.7.1 Setup

- Parameters:

  1. Activation Function:
     We have the 47 features as input and 3 classes as possible output. As the response variable is not binary but ordinal with 3 levels, we choose ReLU (Rectified Linear Unit) as the active function.
  2. Other Parameters:
     We apply the following parameters grid in the GridSearch with 5-fold cross-validation and 'f1_macro' scoring to tune the hyperparameters in MLPClassifier model.

Parameters	Values	Best Value
hidden_layer_sizes	[(21,), (19,), (17,), (15,),(13,), (11,), (9,), (7,), (5,), (3,), (21,11,), (19,9,), (17,7,), (15,5,),(13,3,)]	(13,)
max_depth	[2,6,8,10,30]	10
solver	['adam', 'sdg', 'lbfs']	adam
alpha	[1e-5, 0.001, 0.01, 0.1, 0.2, 0.4, 1, 10]	1e-05

Table 9: Parameters for MLP

5.7.2 Model Performance Analysis

In general, the MLPClassifier gives an average macro F1-score of 0.76 as the accuracy with a greatly unbalanced confusion matrix.

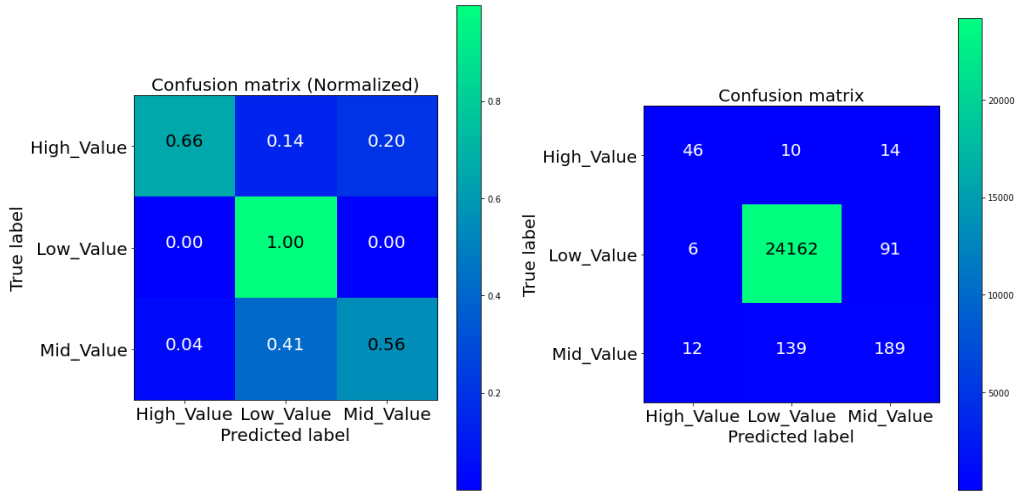
The following classification matrix shows the accuracy of the prediction on 'High-Value', 'Mid-Value' and 'Low-Value'. As mentioned in the previous section, we do not have enough samples for High-Value and Mid-Value donors. As our main goal is to value the High-Value prediction which is the minority class, we should consider a way to smooth the accuracy that might be led by the majority class which is 'Low-Value'. Thus we evaluate the model with macro average instead of micro average which is a weighted average.

As the results, the overall macro average of f1-score is 0.76 with 0.69 of High-Value's, 0.60 of Mid-Value's and 0.99 of Low-Value's. The class 'Low_Value' has extremely high f1-score while the other two classes, 'High_Value' and 'Mid_Value', have much lower ones, indicating the model has better performance in predicting the Low-Value than the other two. And as the 'support' result shows that, in the test samples, There are only 0.3% of High-Values and 1.4% of Mid-Value being predicted.

	Precision	Recall	F1-score	support
Low-Value	0.99	1.00	0.99	24249
Mid-Value	0.64	0.56	0.60	340
High-Value	0.72	0.66	0.69	70
accuracy			0.99	24669
macro avg	0.79	0.74	0.76	24669
weighted avg	0.99	0.99	0.99	24669

Table 10: Classification Report for MLP Classifier

It was a great challenge to improve the accuracy of prediction on the High-Value donor in such an imbalanced dataset. From the following confusion matrix, we learnt that 66% of the true High-value donor in the test sample are correctly predicted while almost all the true Low-Value donor are correctly predicted.



(a) (b)

Figure 19: Confusion matrix of MLPClassifier

5.7.3 Discussion

At last, we didn't find an excellent MLP classifier to achieve the prediction on the value of the donor. One important reason is that MLP is quite sensitive to feature scaling. A little bit change of the features size would lead to different validation accuracy. So the key to generate a well-perform MLP classifier largely depends on the features construction.

6 Challenges

6.1 Imbalanced Response

The problem of inducing classifiers over imbalanced datasets with asymmetric costs has been noted as key open problem in data mining [6].In our use-case with the provided Nationbuilder dataset, it can be noted that the ratio of entries for low-impact customer is substantially high($> 99\%$) when compared to the high-impact customers which makes the result-set imbalanced. As we are trying to predict the tags that would potentially indicate a high impact donator, the imbalance in the resultset might affect the end predictions.

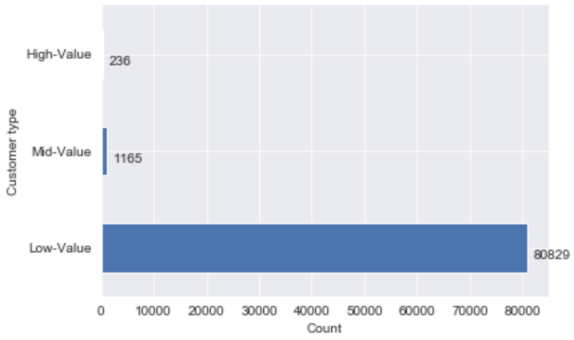


Figure 20: Imbalanced Response

To overcome this problem, the modelling is performed using boosting in the nationbuilder dataset and the accuracy is measured class-wise to denote an unbiased measurement. The F1 macro score of each class is used as an measuring technique and focused on the high-value customers prediction accuracy.

6.2 Sparse Features

Missing data or incomplete data are very common in statistical situations [1]. It is a serious problem because they can lead bias and inefficiency in estimating the quantities of interest. Boosting is one of the most powerful learning ideas that combines the outputs of many weak classifiers to produce a powerful committee. The process is to fit the weaker learners iteratively and reweight the data after each iteration.[12] After performing exploratory data analysis on the provided datasets, it was quite clear that many potentially good features had many missing values and including them in the model will result in bias. For example, demographic data is one of the most essential data used in prediction of recurrent customers, but in the provided data, we can find high missing values in most of the demographic data. So Nationbuilder taglist data was used as the main feature where each tag was passed as feature in one-hot encoded format.

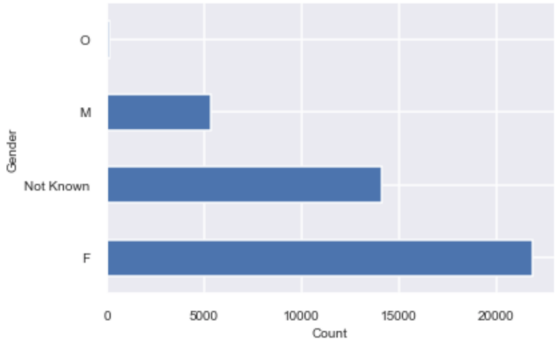


Figure 21: Missing Values - Gender

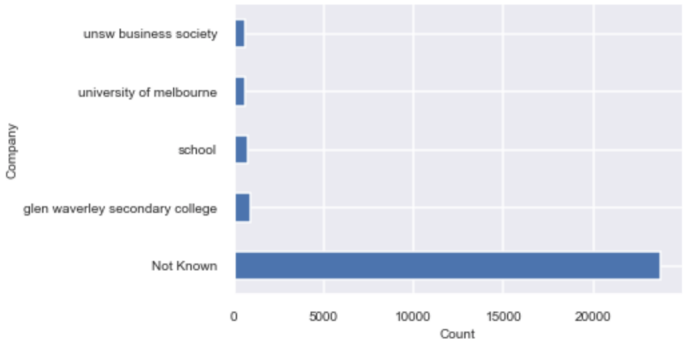


Figure 22: Missing Values - Company

The diagram is projected to showcase the ratio of missing values in potentially main features such as gender, company and location. Missing data or incomplete data are very common in statistical situations. Implementing boosting model to predict the features is one of the aspects that was implemented to obtain better results in predicting the recurrent donators as boosting algorithms handle missing data and imbalanced response better. Another interesting approach to produce quality features is by creating characteristic attributes, instead of trying to impute the missing values in the given feature set. With the NationBuilder dataset, classifying the donators using the recency,frequency and monetary features was one of the prospective options. And also, the tag_list column could be used to produce a One-Hot encoded version of the tags for each donator to create the feature space. This helped in building an unbiased feature extraction.

6.3 Social Media Data Analysis

Identifying and quantifying the factors that influence engagement in social media marketing demonstrates how data analytics can create business value for marketing organizations.
One of the key challenges we faced with the social media analysis was that there was no key available to join the customers with their social media account. Since there is no certain way to connect the two datasets, Nationbuilder and the unstructured social media data, it was not possible to create any possible prediction models. The nationbuilder dataset contains an attribute name_id, which is also encoded for privacy reasons, resulting in no appropriate way to perform the social media analysis.

7 Conclusion & Recommendation

7.1 Conclusion

In summary, throughout the project, we were able to deliver the two requirements of our client:

1. Develop tools (e.g. scripts, dashboards, applications) to better manage/monitor/analyse our data.
2. identify strategies to increase donations to Oaktree. For example, this could involve identifying the characteristics of people who raise most of the money for Live Below the Line.

By having a cleaning R shiny dashboard, the client can save time performing tedious data preparation tasks before any further analysis. Since data quality is a major problem, the tool becomes even more important than any modeling technique..

We also managed to segment the donors into different groups based on the RFM features using unsupervised K-means model. This will help the client develop appropriate strategies for each customer group.

Finally, the best predictive model used XGBoost algorithm to predict whether a customer is of High, Medium or Low value based on the tags they have. Recommendations derived from the result of our model is given in the next section.

7.2 Recommendation

In the end of the project, it is still hard to give specific recommendations on the marketing strategies largely because of the quality of the data. However, we are able to give some recommendation based on the data analysis results and the challenges that we faced during the project.

As we found that "Petition in 2014's End Poverty Event", "Deeply Engaged in Campaigns from 2010 to 2016", and "Is a Donor" are the main aspects that are the most associated with the value segments of a donor, our recommendations are formed as following:

1. Identify the petitioners in 2014's End Poverty event and whether they are still donors now.
2. Stress 'Poverty' in the theme of an event to attract more attention.
3. Identify similarity between campaigns from 2010 to 2016.
4. Explore the distribution of participants in campaign from 2010 to 2016.
5. Give full play to the donors' influence.
6. Identify the retention of the donors' donation or engagement.

As we have clearly described in the section 6, the challenges in the data processing and modelling part of the project are mainly imbalance response, sparse features and social data analysis. To overcome these challenges, we strongly recommend our client to proceed a much more efficient and useful data analysis system by starting from:

1. Integrated Data Warehouse.
2. Improving Collection of Demographic Information.
3. Introducing New Tags.

References

[1] Liang Ye. "A machine learning approach to fundraising success in higher education". PhD thesis. 2017.

[2] Wesley Lindahl and Chris Winship. "Predictive models for annual fundraising and major gift fundraising". In: *Nonprofit Management and Leadership* 3 (Sept. 1992), pp. 43–64. DOI: 10.1002/nml.4130030105.

[3] Ya-Han Hu and Tzu-Wei Yeh. "Discovering valuable frequent patterns based on RFM analysis without customer identification information". In: *Knowledge-Based Systems* 61 (2014), pp. 76–88. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2014.02.009. URL: http://www.sciencedirect.com/science/article/pii/S0950705114000586.

[4] B. Yap et al. "An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets". In: *DaEng*. 2013.

[5] Dong-Sheng Cao et al. "The boosting: A new idea of building models". In: *Chemometrics and Intelligent Laboratory Systems* 100.1 (2010), pp. 1–11. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2009.09.002. URL: http://www.sciencedirect.com/science/article/pii/S0169743909001701.

[6] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. "On the Stratification of Multi-label Data". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Dimitrios Gunopulos et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145–158. ISBN: 978-3-642-23808-6.

[7] Juri Opitz and Sebastian Burst. *Macro F1 and Macro F1*. 2019. arXiv: 1911.03347 [cs.LG].

[8] Thomas E. Torlay L. Perrone-Bertolotti M. "Machine learning–XGBoost analysis of language networks to classify patients with epilepsy". In: (2017). DOI: https://doi.org/10.1007/s40708-017-0065-7.

[9] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *Annals of Statistics* 29 (2000), pp. 1189–1232.

[10] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 3149–3157. ISBN: 9781510860964.

[11] Tianqi Chen and Carlos Guestrin. "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016). DOI: 10.1145/2939672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[12] Robert Schapire. "The Boosting Approach to Machine Learning: An Overview". In: *Nonlin. Estimat. Classif. Lect. Notes Stat* 171 (Jan. 2002). DOI: 10.1007/978-0-387-21579-2_9.

Appendices

1 Github Repository Logs

Github user names:-

- **ASOUVB**:Asouv Bahl
- **Trung Nguyen**: Minh Trung Nguyen
- **cici**: Tingqian Wang
- **fuyaozhang**: Fuyao Zhang
- **Harita Bala**: Harita Bala B

commit 704d03ad5a05917053172f77bb9dfe8a4819dfea
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Sun Nov 1 17:44:37 2020 +1100
Update README.md
commit 3d4328437a74aa3f25d695ccaa933b51ed3d16ea
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Sun Nov 1 17:44:10 2020 +1100
Update README.md
commit 13384e60ac2daa20b2d9710465415561731a00d1
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Sun Nov 1 17:43:44 2020 +1100
Update README.md
commit 89f3f28e7d9f2d29d8ffead26a72695b0298a27b
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Sun Nov 1 17:43:16 2020 +1100
Update README.md
commit dda9dd2be3a4b66a021771f2ea3b5086b05ec5e8
Author: ASOUVB <50261615+ASOUVB@users.noreply.github.com>
Date: Sun Nov 1 17:12:50 2020 +1100
Add files via upload
Meeting Summaries
commit 6f60bf50ee2c85a3b6e10d7e3b6bea05e38342a0
Author: Asouv Bahl <asouv2895@gmail.com>
Date: Sat Oct 24 15:39:23 2020 +1100
Updating for modelling
commit e2717942f0f24d7e83e8847b6ae6012dd00aef62
Author: Asouv Bahl <asouv2895@gmail.com>

Date: Sat Oct 24 15:08:30 2020 +1100
Report Figures
commit 6d38e1857918d0485ba2e6ded0009c67f4154cf9
Author: Asouv Bahl <asouv2895@gmail.com>
Date: Sat Oct 24 14:59:54 2020 +1100
XGBoost and Random Forest
commit d2bbfe71e5251c33432fd6cf11a515fadb9b9432
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 22:37:45 2020 +1100
remove error message for None
commit 0b14f94de432ba97f95004ee16f072083cec2b8d
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 22:29:26 2020 +1100
Update retentionModule.R
commit 3a2da191231c465b2cbafaea68e54ec52ff455b5
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 22:28:41 2020 +1100
Update correlationModule.R
commit 8e098c3efc8709fcd463ab2b2dcb66e861fb261e
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 22:28:01 2020 +1100
Update clusterModule.R
commit 6fb1a74a2cd8963f4bbd46982de28f032c74b64a
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 21:59:18 2020 +1100
Loading symbol - exploration
commit 2e2b1a1deb7dda746d82c870213d8bc5d4252dfd
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 21:57:31 2020 +1100
UI theme changes
commit efce2abda1a961dd9396bc2a03fa692d0b07dd75
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 21:55:41 2020 +1100
UI
commit 18d16aed3ad4a182c3b39b812ab2f20b93ec857d
Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 14:28:03 2020 +1100
Add files via upload
R version of data cleaning script
commit 650bff906d546061fc3df81d39be335f248d7158

Author: Harita Bala <31405532+haritabala@users.noreply.github.com>
Date: Thu Oct 22 14:22:25 2020 +1100
Add files via upload
commit a1664f24d3bc7817f65a2c10d02cf713801927c2
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Fri Oct 9 15:00:01 2020 +1100
A Flow Chart of the Project
commit aaf92d5e2ff9fe8bd324b6520c49385e54fd04e7
Author: cici <46474909+ttcici@users.noreply.github.com>
Date: Fri Oct 9 14:58:41 2020 +1100
Add files via upload
commit f7ce0876fe55d6c858fe0f6ec5ec414df7532199
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 21:02:06 2020 +1000
Add files via upload
Tuning hyperparameters
commit 9c575fcaac17feef221e0225c7a21663e0aa1252
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 21:01:10 2020 +1000
Add files via upload
Decision Tree, GBClassifier and LGBMClassifier
after tuned
commit dea83c4163e5710befd75a8ae07d4cf45b4b040e
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 20:59:51 2020 +1000
Create Predescribe
commit 19317b4a5afcba3d4063a8462b0e8467888120a2
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 20:44:21 2020 +1000
first version of EDA for Nationbuilder dataset
commit dcb47967e30e62870e027e0c7a826a107355bc3c
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 20:35:20 2020 +1000
Add files via upload
commit cf6d053dc9d55bac971ac167784b7021128c9186
Author: fuyaozhang <70028283+fuyaozhang@users.noreply.github.com>
Date: Sat Sep 12 10:26:42 2020 +0000
Create Modelling
commit 366467948b64c3c17d3f42ed11df559c35c9ef1c
Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Sep 2 17:32:23 2020 +1000

labels creation for supervised tasks notebook

commit 109722e8420a10b9a6a39de177249be66317a54f

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Sep 2 17:30:51 2020 +1000

figures for report

commit 6a42cc38c74f91d77bd56e0bd05db5b54117ecba

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Sep 2 17:29:34 2020 +1000

refine correlation module

commit 388179e9102d2e375488649da4a55e06b9279ba7

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Tue Aug 25 11:44:28 2020 +1000

fix date key error

commit ea7472c24ed5576877f63979fa61d699d232327b

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Fri Aug 21 20:53:10 2020 +1000

expand Fix Date to choose which date columns to fix

commit b6eb180eb2904956949483999ceeef0e6b1853f7

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Fri Aug 21 19:51:34 2020 +1000

update clusterModule to 5 clusters

commit b0dd331a75322dec0d0464cc76fe9c47f29e662c

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Fri Aug 21 16:48:45 2020 +1000

refine filter for categorical columns

commit 148b8fbe90d64106a484bf124dc49dde751514e5

Author: Asouv Bahl <asouv2895@gmail.com>

Date: Thu Aug 20 18:37:41 2020 +1000

Adding Select All and Deselect all option

commit 6aa3ede28ce471d7e6ef042c32180c424e0e898d

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Aug 19 19:37:41 2020 +1000

fix taglist indentation

commit ad6d4ceed4ea3cf872ebbfe36dac896ba8b0bf9b

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Aug 19 17:14:47 2020 +1000

convert character columns to factor for the filter to display values and be able to filter on multiple values

commit a3d69a84c704754878b4b27a007723d6c195fd21

Merge: f9b0f62 84e6bd9
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Wed Aug 19 16:01:50 2020 +1000
Merge branch 'master' of https://github.com/nghetrung/oaktree-g14
commit f9b0f627abdbb555899379daa00ea3559110fcfd
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Wed Aug 19 16:01:31 2020 +1000
added options to uppercase the Sex column
commit 84e6bd994512ca4e4351c5c21ed51c991a999876
Author: Asouv Bahl <asouv2895@gmail.com>
Date: Wed Aug 19 15:40:15 2020 +1000
Making dataTable filter downloadable
commit 186a6a12b70cd1189a9218274c9ecdebc05fd927
Author: Asouv Bahl <asouv2895@gmail.com>
Date: Tue Aug 18 23:30:01 2020 +1000
Adding Renaming Functionality
commit 73d8daf92afac26618f9ebb52bbf963c0d81acb9
Merge: 05f05b9 9c41f23
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Tue Aug 18 15:14:06 2020 +1000
added filter for dataTable
commit 05f05b92aa183e03a1647e198b47a166f9138a47
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Tue Aug 18 14:52:04 2020 +1000
added filter for dataTable
commit 9c41f236dd56a929a62fd9aa92201839394d7a2d
Author: Asouv Bahl <asouv2895@gmail.com>
Date: Tue Aug 18 12:48:47 2020 +1000
Adding warning for taglist separation
commit 2ef5785fb2244e52cb19410109189ca811004094
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Mon Aug 17 19:22:25 2020 +1000
added Group By option for Cleaning module
commit 1c6198822bfef057f806a344f405733119fc1e78
Author: Trung Nguyen <nghetrung@gmail.com>
Date: Mon Aug 17 17:13:30 2020 +1000
replace checkbox input by select input for cleaning module
commit f9416f7109d525c81af2de843bd096f531fa5981
Merge: ad310ff b2e57ea
Author: Trung Nguyen <nghetrung@gmail.com>

Date: Sun Aug 16 20:23:26 2020 +1000

Merge branch 'master' of https://github.com/nghetrung/oaktree-g14

commit ad310ff8ed7317aae3ea6402785830f2bb3d9488

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Sun Aug 16 20:21:42 2020 +1000

remove the record with missing ID=-999

commit b2e57ea60ff0afd9c34e1b1f148f13e0179c4008

Author: ASOUVB <50261615+ASOUVB@users.noreply.github.com>

Date: Wed Aug 12 20:04:58 2020 +1000

Rfm features for funraisin

commit d8d72e2b22ff27513eb4b7a04cdcf9a13a30243b

Author: Asouv Bahl <asouv2895@gmail.com>

Date: Wed Aug 12 18:07:32 2020 +1000

Updating For EDA packages

commit 79f0d9ac97e97386e9174ac4e03997c20730c584

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:12:07 2020 +1000

Add files via upload

commit deceb19b674f21cf14877e134ba888fc9f9cd7ea

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:10:46 2020 +1000

Create visualization

commit a2fc3f7feb9f3deee6db180d35b494bdf7994b76

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:08:56 2020 +1000

The first and second EDA

23/4/2020:

title: EDA on Funraisin dataset

issuse: 1. Some attributes that might be useful has over 902. Hard to explore on the ambiguous attributes

25/4/2020(Last Updated):

title: EDA on nanremoved$_{f}unraisindataset$

modification: Data was cleaned again (by Asouv)

commit 74887b21220c427f64247d0c73e95e941734cf92

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:06:30 2020 +1000

Brief Introduction

commit 533a80e3242872155c56e1364240891d76fa2ac3

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:03:00 2020 +1000

the last edition on EDA

commit 408060eb27fc83b08a4a439ea8a25f9889ad57a7

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 17:01:00 2020 +1000

Brief Introduction

commit e2d3fd9fef2332a9b43197763cc09d51045fbbac

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 16:43:12 2020 +1000

Revert "Add files via upload"

This reverts commit 09b72c2e6df2ca9a64af769aa3e027d48f8ceaec.

commit 09b72c2e6df2ca9a64af769aa3e027d48f8ceaec

Author: cici <46474909+ttcici@users.noreply.github.com>

Date: Tue Aug 11 16:34:09 2020 +1000

Add files via upload

commit 15275554f59fcc71b4e2bf471f1cc559e8786433

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Sat Aug 8 22:50:37 2020 +1000

modularized dashboard

commit bbb712128a2a8011f5d2afd5c6b1ff5504b0ec96

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Thu Aug 6 20:14:24 2020 +1000

dashboard 1

commit 174dd9fa79b1dcf4802d8470f02a25f4bcc76fde

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Thu Aug 6 19:46:19 2020 +1000

updated requirements

commit 1339126c83bb473f09f1753f25d4452a67f32747

Author: haritabala <haritabala96@gmail.com>

Date: Mon Apr 27 15:59:20 2020 +1000

KNNCluster fundraising label donator recurrence

fundraising KNN clustering model to predict/label whether a donator is a recurring donator.

commit ba5b01cdad4c8f69c649121c8e75766d805bbcd0

Merge: ab51f29 b564b27

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Sun Apr 26 00:28:32 2020 +1000

Recency, Frequency, Monetary features for nationBuilder

commit ab51f29b2bb5fe1283676d1eddbe3b10021b5058

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Sun Apr 26 00:23:15 2020 +1000

Recency, Frequency, Monetary features for nationBuilder

commit b564b2770ce5bd5a34c1207198c4fae49ea2f979

Author: asouv bahl <asouv2895@gmail.com>

Date: Fri Apr 24 17:24:06 2020 +1000

Social Media Feature Engineering

commit bc0d62e2fd36fe816350f5d0b28d09ee4521f3e6

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Thu Apr 23 14:36:36 2020 +1000

merge facebook files

commit 4007e5f11759b2790efae65e5cddced2e9fbbaa1

Merge: b41064d 8327f79

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Mon Apr 20 21:58:15 2020 +1000

Merge branch 'master' of https://github.com/nghetrung/oaktree-g14

commit b41064d3347295aaf05a0ea64af9ecab96c336a3

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Mon Apr 20 21:56:15 2020 +1000

merge nation builder data and perform initial cleaning such as removing many nulls columns and separating tags

commit 8327f79059b7ec25dbf270b9a08e0509035207a6

Author: asouv bahl <asouv2895@gmail.com>

Date: Fri Apr 17 21:52:00 2020 +1000

Funraisin cleaning attempt

commit 64ed4ca1069d50abb8f24af3bc417c2ef626a846

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Apr 1 13:55:42 2020 +1100

Rename cleaning notebooks

commit 8020c5af546154f94bbdfd41a6bf7f119e0b2a21

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Wed Apr 1 13:48:18 2020 +1100

Initial cleaning and merging NationBuilder data

commit cab0b31955ad0a14d7e8fca208ba302a9a5b5c43

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Mon Mar 30 14:20:11 2020 +1100

First cleaning attempt on Funraisin data

commit 4d308b05fe0734edcbab6d6f0a98494f251e2164

Author: Trung Nguyen <nghetrung@gmail.com>

Date: Tue Mar 24 18:05:44 2020 +1100

Create README.md

commit bc6c7fb6fda84e51127805b5f1c5ac90b7d9f652

Author: Trung Nguyen <nghetrung@gmail.com>
Date: Tue Mar 24 17:59:42 2020 +1100
first commit

2 Meeting Summary 27^{th} March 2020

Opening:
First Client meeting with Oaktree for the industry project 2020 to get an overview of the entire project. This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 27 March 2020 at 2:00PM AEDT.

Present:
Vivek Katial vkatial@student.unimelb.edu.au Project Supervisor,
Michael Kirley mkirley@unimelb.edu.au Coordinator,
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- Overview of the company and project by Thomas
- Various projects held by the company
  - Funding to international organisations
  - Youth participation
  - Student ambassadors workshops
  - Policy and advocacy by younger Australians
- Fundraising
  - Live Below the Line
  - Previously Peer-to-Peer Marketing
  - Smaller Campaigns
- For more detailed understanding. Websites:
- https://www.oaktree.org
- https://www.livebelowtheline.com.au
- Marketing Strategies
  - Facebook Ads – last year – Not sure if we have complete access to this data yet
- Expected Outcome 3 – Dashboards – Social media activity overview– posts and likes – Weekly and Daily report on donations which are made directly through the website
- Initial set of data can be accessed through OneDrive which will be provided by Thomas by this weekend.
- Google Analytics Data would be provided to us
- Nation Builder Dashboard Data API – Thomas will confirm with us regarding the access once legal student deeds are provided.
- Duplicate record issue currently being fixed by Thomas
- Facebook API Data could be useful for the analysis as well
- Business view of the data – volunteers are more responsive – supporter databases – live below the line website DB – Donations DB – Oaktree website Database – the formats and fields might be upgraded over the years – information overlap between the databases could be found

Adjournment:
Sample Dataset will be provided by the client within this week to get a brief idea to understand what we will be working. We will schedule another meeting with Thomas within the next three weeks.

Minutes submitted by: Harita Bala B

3 Meeting Summary 18^{th} April 2020

Opening:
Second Client meeting with Oaktree for the industry project 2020 for better understanding of the data. This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 18 April 2020 at 2:00PM AEDT.

Present:
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- Instagram and Twitter Access given by account details - Use only to access data.
- Google Analytics and Facebook access given.
- Facebook Ads from third party marketing company from last year can be accessed if required.
- Facebook – analyst role. LinkedIn- full access.
- No complete access to Nation Builder for legal reasons – Dashboards analysis if necessary can be created on social media posts data.
- NationBuilder_id is different keys and cannot be used to join the databases "People" – user_id and "Donations" – transaction_id
- Missing Data values(though compulsory in website) can be because few fields are manually updated from Live Below the Line campaigns.
- "tag_list" – most important column
- Name_id is generated based on letters and doesn't give certainty that its different users.
- NaN values other than blank values – reason will be sorted out by Thomas later.
- Created_time and all timestamp field have formatting issues which has to be fixed before analysis in NationBuilder dataset. Formula to fix is available in the provided dataset excel.
- Merging the different dataset will have challenges.
- Description on each field on every provided dataset was explained by Thomas.
- Rare manual deletion of teams in Fundraisin – teams can be found due to legal data privacy concerns raised by endusers.

Adjournment:
Email with all the descriptions of fields, abbreviations in the "tag_list" and clarifications will be provided by Thomas. An entire understanding of the datasets provided was gained through this meeting. Access for all the social media account to the team will be finalised by this week. We will schedule another meeting with Thomas in the next three weeks to discuss the progress and clarifications on the project.

Minutes submitted by: Harita Bala B

4 Meeting Summary 8^{th} May 2020

Opening:
Third Client meeting with Oaktree for the industry project 2020 to update Thomas on the exploratory data analysis and confirm the correctness of the approach taken. This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 8 May 2020 at 3:00PM AEDT.

Present:
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- EDA demonstration by Cici, Trung and Fuyao.
- Cents to dollars in "Donation Amount"
- Fundraising dataset – School, College and University can be merged. Another column to address the presence of anyone of this data can be added(Yes or No)
- Group by Member ID to be done in few calculations in Fundraising EDA to remove duplicates and avoid errors in visualization
- Instagram account will be linked to Facebook to access the API
- Tag_list contains the most effective data in NationBuilder
- Tags with very minimal count in comparison to the distribution of tags can be removed. For eg. Tags which occur only once in the entire dataset can be removed.
- Task – model using the recurring customers to obtain features of potential customers and then label the entire participants based on the features extracted from the model.
- Next task : Social media engagement can be linked to each campaign's timeline and analysis can be done to see if there's any potential useful data there.
- Discussion on different campaign's and their timeline with regards to revenue generated.

Adjournment:
Minor data clarifications will be provided by Thomas. Come up with multiple approaches to solve the task and obtain the list of potential donators. We will schedule another meeting with Thomas in the next three weeks to discuss the progress on the project.

Minutes submitted by: Harita Bala B

5 Meeting Summary 17^{th} July 2020

Opening:
Meeting with Oaktree for the industry project 2020 to update Thomas on the next steps and confirming with him the deliverables.This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 17Jul 2020 at 4:00PM AEDT.

Present:
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- Trung and Asouv presented an initial sample of shiny dashboard to give Thomas an idea of how it will look and to discuss the feasibility of the applications of dashboard.
- Thomas appreciated the initial sample.
- Thomas expressed the need of adding a cleaning module in the dashboard as he was doing the cleaning manually and the task was becoming tedious.
- The team agreed on delivering the dashboard with a cleaning module.
- Modelling expectations discussed and possible challenges discussed promptly by Trung.
- Social media challenges explained to Thomas by Asouv and Harita .

Adjournment:
Thomas would finalize the functionalities Oaktree want in the cleaning module of the dashboard and will send a follow up mail regarding the same. The next meeting will be held on 25th Aug 2020 (Keeping in mind the time required for developing the dashboard and exploring modelling .)

6 Meeting Summary 25^{th} August 2020

Opening:
Meeting with Oaktree for the industry project 2020 to update Thomas on the cleaning module of the dashboard and to confirm if the functionalities met the expectation of Oaktree.This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 25th Aug 2020 at 2:00PM AEDT.

Present:
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- Cleaning module of the dashboard presented to Thomas by Trung and Asouv.
- Remote access to the dashboard was provided to Thomas to provide him with an Hands-on experience.
- Thomas acknowledged that dashboard did meet the needs of Oaktree.
- Approach for modelling – Clustering , creating labels, Feature selection
- Initial results from modelling discussed.

Adjournment:
Cleaning module did suffice the needs of Oaktree . Modelling results will be discussed with the presentation in the next meeting . The day/time for next meeting will be communicated to Thomas.

7 Meeting Summary 16^{th} October 2020

Opening:
Final meeting with Oaktree for the industry project 2020 to show Thomas the results from modelling and get feedback from Thomas on our final presentation. This meeting was organized by Asouv Bahl and team with the Data Analyst of Oaktree Thomas Crawley on 16th Oct 2020 at 4:15PM AEDT.

Present:
Thomas Crawley t.crawley@theoaktree.org Thomas Crawley,
Minh Trung Nguyen trnguyen@student.unimelb.edu.au Team Member,
Fuyao Zhang fuyaoz@student.unimelb.edu.au Team Member,
TINGQIAN Wang tingqianw@student.unimelb.edu.au Team Member,
Harita Bala Balamurali hbalamurali@student.unimelb.edu.au Team Member,
Asouv Bahl asouvb@student.unimelb.edu.au Team Leader

Summary:

- Different algorithms that were used to model the problem were discussed by Tingqian, Fuyao and Asouv.(Why we chose the set of algorithms we used in modelling)
- Results from the algorithms discussed and presented to Thomas(Feature Importance, Result of Macro-averaged F1-score for all the algorithms used.)
- Tinqian presented the recommendations to Thomas- The set of tags that were important for identifying high value donors and ways in which Oaktree can retain these donors and increase their donations by focusing on a particular theme in their campaigns.
- Thomas appreciated the work of the team and gave a good feedback on the final recorded presentation.
- Access to the Github provided to Thomas for accessing the code for modelling and dashboard(Dashboard was not deployed as Thomas wanted to run it locally and as a result the requirements for running the code for dashboard on his local machine were discussed.)
- The team thank Thomas for providing the opportunity and Thomas appreciated the effort of the team throughout the year.

Adjournment:
Access to Github repository will be confirmed by Thomas. A copy of the report will be shared to Thomas.

8 Further Information

If any further details(including access to Github repository) are required then please contact us through the emails provided in the title page.