City, University of London. MSC in Data Science

Project Report - 2021

Explainable Clustering as an Approach to Life Insurance Customer Segmentation

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed: 

Dominic Palaczky Student Number: 170011271

Supervised by Aidan Slingsby

21/12/2021

Abstract

Customer segmentation is widely used industry practice for identifying groups of customers that share similar traits, through the analysis of these groups it is possible to gain an understanding of the underlying behaviours. By the use of clustering, this project is aimed at providing an approach to customer segmentation that is unbiased and explainable. Through the use of these explanations it is demonstrated how it is possible to produce an automated customer segmentation approach that can give insight into the distributions of customer segments. In addition, this project explores the use of customer segmentation to give insight into unknowns within the data. The approaches are explored through the lens of the insurance industry and Gone-Aways, providing a real use case of the methods proposed.

1. Introduction and Objectives

1.1. Background

Customer segmentation is an enduring approach to consumer analytics that has been applied most widely to retail applications where the data on consumers is readily available. Physical retail stores naturally evolved to adapt their products to suit the needs and wants of their local customers. As time passed, data on consumers became more widely available, for example there was an understanding on how shop layouts can affect behaviour of different groups. It is also in retail that the approach of clustering for the purpose of customer segmentation is most widely used, and different approaches have been explored and developed.

Recently, this approach has been adapted to other sectors, as an approach to modelling consumer behaviour and gaining deeper insights. The uses are less obvious, but just as wide reaching as that in retail, as an example, if it can be understood what segment of your customer base is most likely to respond to a certain type of advertising modern web advertisers can target that group. It can be argued that more traditional approaches of segmentation can be confusing, unrepresentative or discriminatory, and as such an explainable, unbiased machine learning method should be developed.

Clustering as a machine learning approach is well established, and is almost synonymous with unsupervised learning. Many different approaches have been used, some taking heavy inspiration from statistical approaches, some from geometry and others from more abstract neural network ideas. The idea is to model features of the client base in the clustering algorithm, and then take the output clusters as distinct groups of customers. This can be performed on simple characteristics, such as sex, marital status, occupation or on behaviour based features, for example the amount of money spent, or the number of a certain item purchased.

Despite the wealth of research in the approaches and algorithms for clustering, as well as limited research into how to explain the clustering, there is little concrete guidance on the best approaches. This is explored in the literature section of this report, to summarise the current approaches are primarily ad-hoc.

This project focuses on the insurance industry, which in the UK has been quite technologically advanced in a lot of areas. The most common use for machine

learning in this sector is price determination, which is relatively simple. In the case of motor insurance, many details of an individual and their vehicle can confidently predict the likelihood of an accident and in turn determine a quote. This, in essence, proves the suitability of machine learning approaches to the insurance industry.

In particular, this project focuses on Child Trust Funds (CTF). These were a long-term tax-free savings account for children, that reaches maturity when the child turns 18. The first accounts were created for children born in 2003, and as such they have begun to reach maturity. This has highlighted the issue of Gone-Aways, as many have not claimed their accounts. A Gone-Away, also known as a dormant, is identified by an individual who has either left or forgotten their CTF account, but it remains open. This means the account has a value, but has not been interacted with recently. It is estimated that the value of the accounts of these gone-aways is as much as £3 billion, showing the full extent of this problem.

This project is being conducted as part of an internship for Ai-London and as such is guided by what the client and in turn Ai-London desire and recommend. There are certain approaches and models that have been recommended to tackle this problem, but the research, methodology, and results will be of my own work.

1.2. Data

The data for this project will primarily come from Kingston Unity. This was a life insurance mutual company that prides itself on its social responsibility, as a mutual their members in part own the business. In the case of gone-aways this is important, as there is little incentive for the average company to encourage individuals to withdraw their money from accounts; a company's profits are directly related to the value of their liabilities. This can have an impact on the meaning of the results, so this research project will focus on a transferable methodology.

However, due to confidentiality rules, the data submitted with this project is from a mock dataset, which will be referred to as 'dummy data', generated following the distributions that exist within the original dataset, as discussed in the methods section. This dummy data set is beneficial, as it allows for the manipulation of data to explore a 'best case' scenario, and will be the primary source. The 'real' data is used to explore whether meaningful results are produced. Further, location data will be

omitted from any clustering segmentation. This erases any ethics issues with working with real identifiable data, and working with data of individuals who may be under 18. In the results section, the methods will be applied to the real data. This was performed following external regulations, again with identifiable information purposefully omitted.

1.3. Research Questions

The research for this project is designed to guide the development of the automated clustering as desired as an outcome, and is a proof of concept for such a system to exist. Based on experience gained through the internship at Ai-London and experience in machine learning and clustering the following questions are proposed:

What clustering techniques can be applied to make customer segmentation based on human behaviours with insurance accounts through explainable clustering?

How can clustering and customer segmentation be useful in providing the likelihood of identifying Gone-Away accounts?

Can an automatic approach be developed to solve the problem of customer segmentation and provide meaningful insights?

1.4. Objectives / Outcomes

The objective of the clustering is to produce meaningful groups as it pertains to CTF Gone-Aways in this context. It is an aim of this project to produce a systemic approach that can be replicated in many different use cases for customer segmentation. The approach should encompass the clustering, the explainability and the labelling of data.

For the beneficiary of this project, Ai-London, it is important to produce a working script that can automate the process of customer segmentation. The design of which is to be fed a dataset, and information on what should be labelled and output the customer segments, an explanation and visualisations. This can be run internally on demand, to understand data better, or externally by clients, to make policy decisions within insurance companies. These policy decisions would be, for example, to

encourage behaviour of policyholders as to prevent them from being Gone-Away, as exhibited by the non Gone-Away accounts.

1.5. Work Plan

The work for this project was in part work with an internship taken at Ai-London. Initially three months, the initial proposal for the project at this stage was to create a recommendation model for the best tracing service to identify an individual. Clustering was to be utilised as the Gone-Away label in the data was not definitive, and such an unsupervised approach to segmenting customers with the purpose of understanding behaviour and in turn identifying those likely to be Gone-Aways. The bulk of the work conducted in this period focused on learning about the insurance industry, the data and exploring the phenomena of Gone-Aways, extracting information and discovering how to identify and trace them. It was found towards the end of this period that the tracing services were slow to get working with and therefore unreliable for a project of this nature.

So, the project has developed to focus more on the clustering and customer segmentation. This was aided by an extension to the internship at Ai-London for three further months. This allowed for less reliance on external sources for the systems to work, and for a deeper dive into the clustering, allowing for more research to be conducted on visualisations and explainability. The second three months focused on this, with the development of the automated system towards the end of the project's timeline.

1.6. Structure of Report

Following this introduction, this report will explore the critical context surrounding this project. In the context section, the current state of clustering, particularly for customer segmentation will be examined, and the visualisations used. Of interest are the techniques and markers that are used in the literature to determine the best course of action for that case.

The data used for this project will be explained, the steps taken to obfuscate personal information as well as how an artificial dataset was produced. These datasets will be analysed.

Next, the report will focus on methodology. Initially in this section it will highlight the work done on developing a traceability score for the policy holders, this is relevant to the problem of gone-aways. It will then transition to how a script was developed to automatically cluster and explain customer segmentation. There is also a walk through example of how the methods will be implemented.

In the results section, the effectiveness of clustering as an approach will be explored. The tests proposed will be used to analyse the methodology. This section will focus on the performance of the machine learning algorithms, as well as analyse the performance of the explainable clustering attempts. To test the performance of the approaches, the results of the script designed here will also be tested.

This report will then discuss the methodology, through the quality of the results and comment on how this work has benefited the current body of research.

There is a reflection at the end of this report. The reflection section discusses personal thoughts on the project as a whole as well as each section individually. Looking at the decisions made throughout the report with a critical lens.

The appendix for this report contains a glossary and then two small reports on different parts of this project which are not directly related to the research, but are interesting in their own merit. The first is an explanation of how the traceability score was calculated, which was the original focus for a dissertation. The second is a discussion of how the dummy dataset was created, and of particular interest is how and why this dataset was edited and manipulated to benefit the development of this project. Also in the appendix is a collection of code snippets that exemplify the work done throughout this project.

For a full collection of code and results there are jupyter notebooks compiled to show the full process of the project's programming.

2. Context

The research for this project is neatly segmented due to the nature of the project. There is the research into the clustering approaches, where clustering on account of behaviour but with a

goal of visualisation was focused on. There is also a wealth of literature on customer segmentation, some of which relates to clustering which is used primarily to determine the best approach to feature selection. Finally studies into explainable AI and in particular clustering will be explored.

2.1. Clustering

The most widely used algorithm for the purpose of clustering is K-Means, which can be seen as a baseline as it is easy to implement and proven effective . There have been many adaptations to the algorithm with the aim of improving computational efficiency like global K-means (Likas et al.). This approach performs well like in a complex system as is being modelled in this project due to the deterministic nature of the algorithm, but will not be used for this project as there is little need to focus on optimisation. Also there are attempts to improve the finding of the centroids (Zhao et al., 2018), but again focuses on the development of a more efficient algorithm, this time using an explicit objective function. An adaptation to consider here is that of K-means vulnerability to high dimensionality, so an approach like PCA could prove useful, particularly in visualisations (Ding & He, 2004). PCA transformations are not unique to the K-means algorithm, and other transformations like TSNE can be more meaningful (Van der Maaten & Hinton, 2008). This paper highlights how TSNE is used in visualisations to preserve patterns between data points, and will be useful in visualising our K-means approach.

For K-means there have been many different approaches proposed to determine the best value for ‘K’, and exploring them can require research of its own (Kodinariya & Prashant, 2013). The elbow method is a simple but effective way that through a visualisation the number of clusters can be determined. A way to detect this point systematically can be to see it as a ‘knee’ point which allows for an algorithm to be produced to identify the location of this point (Satopaa et al., 2011). Alternatively, another popular approach is to create silhouette plots, which is also a useful approach to cluster visualisation to determine the ideal number of clusters, and can also be used to validate the clustering (Wang et al., 2017), (Thinsungnoena et al., 2015). Alternatively a statistical approach could be used as Hamerly did, applying a hierarchical approach to K-means to test a hypothesis (Hamerly & Elkan, 2003).. This

approach is less appropriate for this project as it does not provide a detailed visualisation as the others do.

Alternatively, a neural network approach such as a self-organising map (SOM) is particularly suited to visualisation problems (Kohonen, 1990). Kohenen proposed that data can be modelled by neurons, adjusting the weightings according to similar data points and this allows for the neurons to organise the data neatly into groups of similar items. If necessary, a SOM can be adapted to handle mixed data types including categorical, by adjusting the weightings for the classification as the algorithm runs. A further alteration to the classic Kohonen SOM is to remove the constraint of a fix map size, which allows for a greater projection map (Tai & Hsu, 2012). This is useful for problems where the optimal size of map is not known, but can produce maps that are of very large size. In this approach, the SOM is proposed to be used as a visualisation tool to help understand the decision process, so using this approach could lead to maps that are too large. SOM is widely used as a visualisation tool, like in the paper 'Visual analysis of self-organizing maps' (Stefanovič & Kurasova, 2011), which highlights various strategies for visualisation as well as touching on software that is capable of producing these. This provides a baseline for what is regarded as a 'good' SOM visualisation, with an emphasis for this project on grid size being kept low. In particular, the pie charts which show how much of a class is within a neuron will help in understanding the problem of Gone-Aways.

For financial data, there has been research into how clustering can be used. This survey (Cai et al., 2016) provides an outline, and can be seen as best practises in how to use financial data in clustering analysis. Also of note in this paper is the discovery that density based clustering without alteration is ineffective, suggesting using weighted distances instead (Yang et al., 2020). There is also a suggestion to use classifiers to improve the interpretability of results, which will be used in this project, but explored further in the explainable cluster section. Further, there has been research into the use of SOM particularly for financial data, of note here is the first chapter of Visual Exploration in Finance (Serrano-Cinca, 1998), which gives an overview of the utilities of SOM as it relates to finance. The chapter gives markers for what a successful SOM can look like and how it can be integrated into decision systems. However, for this context in looking at insurance data and a single policy it is only tangentially relevant. Also in the book is a chapter on Chinese markets, and how

it can be used to simplify customer segmentation (Schmitt & Deboeck, 1998). This chapter is evidence that a SOM is useful when it comes to consumer segmentation, but in this research context is limited by the focus on market research.

2.2. Customer Segmentation

Customer segmentation is a well established analysis tool, primarily utilised by marketing teams within retail, but the possibilities have expanded beyond this, and into other sectors. The standard approach is to take a marker, and explore how groupings of customers can affect the retail business, such as in this paper which looks at customer value (Marcus, 1998). In this approach, it is also explored how using this technique has allowed for greater revenue through representation using a customer value matrix, indicating that customer value is the most important focus for this type of clustering. Strategies could then be developed which expand beyond retail, as in this paper, which looks at value analysis and uses wireless communication (Kim et al., 2006). This is notable as a decision tree is used to explore marketing segments and validate clustering proves effective.

Clustering has a natural application in customer segmentation. Using machine learning approaches, more insight can be extracted, the paper (Wu & Lin, 2005) examines how patterns in credit card data can be utilised, and hidden characteristics can be extracted . This paper also produces insight into how feature analysis after clustering can be used to validate the performance of the clustering. Further, there are applications of this within fintech more recently; (Sheikh et al., 2019) this paper performs two-stage clustering including K-means for segmenting customers to decide targets. This validates that K-means generally is a valid approach to clustering financial customers. It also contains a systematic approach that has enabled space to be expanded upon in this research.

Of particular interest in this research is how these classifiers were applied to create explainable segments of customers (Albuquerque et al., 2012). Exploring this deeper, to provide a more advanced method of obtaining clusters from support vector machine to create internally homogenous groups. Using this, the researchers created explainable groups, highlighting the features that make each group unique. From here they highlight how policy changes could affect different groups.

The purpose of clustering in this project is to discover meaningful groups of policyholder behaviour. This is something that has been explored in the telecommunications sector, where it is found that behaviour is an important method to segmentation (Bayer, 2010). Micro-segments are also used heavily, that being specialised groups of customers which have a strong tendency towards a particular feature. Micro-segments provide a more targeted policy approach for the end user. This, in combination with other articles in this section, provide a solid foundation to techniques that can be explored further and applied to this particular use case, but also provide evidence that techniques are transferable between sectors.

2.3. Explainable Clustering

Explainable AI is a wide research area for supervised learning, the paper by Roscher et al. (ROSCHER et al., 2020) goes in depth about the current scientific approaches to tackling this problem. The article bridges the gap between the current state of explainable machine learning and scientific works, but can also be seen as a survey of the current state. In particular, there is a distinction made for interpretability, this meaning the ability to actually make sense of a machine learning model, and explainability, the explanation of how a model has made its decisions. The research also highlights that work needs to be done to integrate the explanations into domain knowledge, translated to this project meaning that it is necessary to explain the meaning of the customers as it pertains to customer segmentation, not just the dataset.
On a higher level, there has also been research into defining the principles for explainable machine learning (Belle & Papantonis, 2021). This paper explains the differences between opaque and transparent models, depending on how obvious the architecture of the model makes the decision process. The models that will be used in this research can be seen to fit in the transparent category, but will use the principles of model agnostic opaque explainability, as this project is designed to be transferable. Simplification will be explored through use of a decision tree, and the feature relevance will be utilised. It is worth noting that this research is not fully transferable, as in this project the nature is unsupervised, without clear labels the exact methods are not transferable.

There is little research for explainable clustering, so the principles of explainable machine learning must be adapted. For K-means, some research has been performed, like utilising a tree structure to explore decision making (Dasgupta et al., 2020). This paper highlights the trade off for tree size between predictive score and interpretability, highlighting a limitation of this approach.

The body of research for this subject is lacking. It is worth noting that research is primarily utilising purely computer science ideas, with little attempt to use statistical approaches. Further, it is worth noting that clustering for customer segmentation use cases will benefit from a robust explainability methodology, as the final user can be assumed to not be a data scientist.

3. Methods

For this project, the methodology developed is to be developed into a script that can be run to automatically segment customers, then have the ability to give insight into these customer segments and finally the Gone-Away accounts. This methodology can also be seen as a systematic approach, one that is justified and can be applied as a ruleset for other customer segmentation problems. The implication being that the methodology here must have a logical path to follow; this must convey meaning and explanations. It is the aim of this section of the report to expose the framework behind the methodology, and provide the exact algorithms and approaches that have been used in producing this body of work.

This chapter will begin by explaining the requirements for the data. It will then be stated the requirements and libraries this approach uses. The design will be explained step by step, with information on how it was implemented on the KU dataset.

3.1. Data Requirements

The data for which this problem is trying to solve will be customer focused, and is preferred to contain features that relate to a customer's behaviour. This is because the current industry solution for customer segmentation is to focus on individual traits, that being items like male/female, single/divorced. Whilst this data is useful, it is also generic, meaning the amount of information to be extracted is surface level. Further, the utility of machine learning to segment customers based on these metrics is limited,

this is shown by the desire to gain a deeper understanding of customers by the insurance industry, and by the beneficiaries of this project.

In this project, behavioural data is a phrase used to describe data that conveys how a customer acts. This is referring to data items like the length of time the policy is held, the frequency a policy information is updated. Use of this data is preferable, as it is more revealing than using data like sex, and assuming that those of the same sex act in the same way. The current method is to state that single males over 40 are more likely to have high value accounts, for example, whereas this methodology is developing a way to say that individuals that update their account information often and pay small but often premiums are better to target in a certain way. This is not to say that customer information is not useful, but it should not be the only metric customer segmentation is performed on.

For this project, the data should be continuous, this is because the clustering and the methods to describe those clustering techniques rely on distance based approaches. Further, the data should not be categorical, again this is due to the distance based algorithms. If the data is categorical, the values should be adjusted to convey euclidean geometry, like this project has done with the sex category.

Adjustments to the data should be made, like the normalisation proposed, and if a feature is right skewed a log value should be taken, but the data should be fed to the clustering algorithm in its entirety. It is the purpose of the clustering to distinguish between similar data points, and as such by, for example, separating a two peak distribution of a feature, should be performed by the clustering algorithm if it is appropriate.

3.2. Data Gathering

The data for this project will be financial centred behaviour. In this project, the focus is on producing a methodology and implementing it as a script . For this reason, it is beneficial that there are multiple datasets, so that the approaches can be tested on different scenarios. The features of the data will be behaviour focused rather than personally identifying. In this chapter, three data sets (one artificial and two real) will be explained and initial analysis on each will be explained.

The primary data for this project is provided through Ai-London from Kingston Unity (KU). The data was limited to just those with CTF accounts, consisting of 90,000 policyholders. This data contains personal, identifiable information, so could only be accessed in the office at Ai-London, on a separate secure computer. This data was then modelled to create a separate 'dummy' dataset.

The structure of the KU data that was accessed was vast, consisting of hundreds of tables individually containing information on the individual, and their policy. The data was accessed through oracle database[1], using SQL queries, which was necessary to pull the data and perform basic filtering and aggregations. The tables were then saved as three separate CSV files. The first contains personal information for an individual, the second contains the address history for each individual, and the third contains transaction histories.

This data contains personal identifiable information, so it must be kept secure and is not able to be submitted for this project. It is for this reason that there was a process of creating a 'dummy' dataset, which shared the features and distributions of the original data set. The dummy data also allowed for manipulation of the data, to allow for a better distribution of the features than there is in the real data, allowing for a 'best case' scenario. This is the data that accompanies the report, and is suitable for the models mentioned in this report to be run on. The code for producing this was done in python, and used random numbers, predominantly following a skewed normal distribution. The data was generated to replicate the tables pulled from the KU database before any preprocessing, and was performed after feature selection occurred, so just contains the best features for this dataset. This allowed for easier replication, and the analysis performed could be unique to this new dataset.

The dummy data was manipulated throughout the design process of this project, this was done so that the methods were not swayed by potentially poor data. To do this manipulating, either the centre of the skewed distribution was adjusted, or in cases where there were twin peaks in the original data, the percentages of these were set to a more equal level.

[1] https://www.oracle.com/uk/database/technologies/

Finally, there has been a third data set gathered from databricks[2], which focuses on the motor insurance industry. This dataset contains 1000 rows focusing on claims, and is designed to be used in predicting fraud detection. This dataset will be used in this project to further verify the transferability of the methodology. This dataset can be seen as segmenting customers based on their claim information, to determine the characteristics of those likely to conduct fraud. This is a much more challenging task, the dataset is small, and the differences between fraud and not fraud cases are notoriously small. So the purpose of assessing this dataset is to see how the approaches perform in a difficult scenario to identify the areas that require more work.

3.3. Features

The features here are for the real and dummy data sets.

From these tables, aggregations were made, meaning averages of the time series data, like mean premium payments. This was to make the inputs to the models represent the whole of the data, following the advice within the literature. The features of the data are:

Perno - a unique integer for each policy holder, used as a index, not feed to ML model
Sex - a categorical data, 0 = male, 0.5 = other, 1 = female
ValueLog - the log of the value of a policyholders account
GoneAway - a binary value given 1 if the account has been manually labelled. This will not be an input into the clustering itself, and can be seen as semi-labelled.
moveFreq - the number of times an individual has moved
moveDur - the average duration of days an individual stayed at an address
premsFrequency - the number of times premium is paid
premsMean - The mean value of a premium paid by account
premsMax - the maximum value of a premium paid by account
premsMin - the minimum value of a premium paid by account
premsMedian - the median value of a premium paid by account
Traceability - the confidence score for the individual to be identified on public records

All features in this section will be normalised. The algorithms used in this report are similarity based, and some are using distances. This means that some larger values, like the maximum premium payment, cannot outweigh smaller features, like sex simply because the difference between them are

[2] https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4954928053318020/1058911316420443/167703932442645/latest.html

naturally larger. Further, the log of certain features, like the value of an account is taken. This is for a similar reason, as the distance between values should be representative of the distribution.

3.3.1. Traceability

Traceability was the initial focus for this project. Here traceability means a score calculated from the likelihood of an individual showing up on public records. As this project transferred away from the problem of actually tracing gone-aways, only a summary of this work will be included in this paper, and the value given will be an input into the clustering models. In summary, an external API was used to search the public records, then a confidence score was calculated, to predict the likelihood of a person being found. This used similarity formulas, weighted by the importance of feature. The API is monetarily expensive, so a predictive model, using a hybrid of classification and regression decision trees was created to predict the confidence score. Calculating this score requires personal information, so this approach could only be applied to the real data. This score will be modelled by the dummy data.

3.3.2. Gone-Aways

Gone-aways, or dorments, are described as individuals who have not interacted or replied to messages about their account. In the KU dataset they are manually labelled, and as such is not a reliable feature. This means, we can trust those that are labelled as gone-aways, but not the accounts that are labelled as not gone-away. The clustering algorithms will not use gone-aways as a feature, rather it will be explored how useful the clustering is at defining these accounts.

All-in-all this project is focusing on methodology, so the dataset for this was tailored to have features that are not entirely unique to it. For this reason, the data is mostly continuous, which makes it easy to process when the ML models are trained. Further, there are two primary data sets that will be discussed through the remainder of this paper, the dummy set, which is used to explain the methods, and the real data which will be discussed in the results. Unless otherwise mentioned, this report will be using the dummy data.

3.4. Data Analysis

In this section, the real KU data used will be analysed. This can be seen as an initial look at the datasets, the distribution and the relations between features. Also in this section, proposals for any adjustments to the dataset will be proposed where the analysis reveals it is relevant.

Table 1 shows the summary statistics for the real data set. It can be seen that there are 93,000 data points in this set. For the sex feature, there were a large number of unidentified entries. This indicates that there is likely a human input error at some point. The effect of this is minimised, as the primary records of interest here are of high importance for the insurance company to be correct, like premium payment records, but should be kept in mind when looking at these specific data points. To mitigate this effect, those data points without a given sex will be given a value equidistant from the male and female categories.

	perno	**sex**	**value**	**Gone Away**	**Move Freq**	**Move Dur**	**Prems Freq**	**Prems Mean**	**Prems Max**	**Prems Min**	**prems Median**	**Prem sDur**
count	93563	93563	93563	93563	93563	93563	93563	93563	93563	93563	93563	93563
mean	106554	1.77362	632.586	0.15609	1.0073	6522.8	2.8247	217.19	230.29	208.03	218.16	2965.1
std	54679.3	0.57210	797.104	0.36295	0.0866	552.34	12.358	216.28	248.99	211.79	216.98	3223.7
min	6063	0	0	0	1	0	1	2.56	37.63	0	1	0
25%	59426	2	376.838	0	1	6570	1	250	250	200	250	30
50%	106683	2	549.761	0	1	6570	2	250	250	250	250	182
75%	153714	2	811.034	0	1	6570	2	250	250	250	250	6570
max	201571	2	42186.6	1	3	6570	922	24792	24792	24792	24792	6570

Also of note in this table, the data will be required to be normalised. The highest value is significantly higher than others. As distances between data points will be used in the clustering, it is important this is not skewed by large values. This can comfortably be done between the values of 0 and 1, as no data is negative.

The features about premium payments are all similar in distributions, with the majority of accounts having just 250 paid into them. This is a reflection of the fact that the government paid this amount into all CTF accounts when they were opened. Over 75% of accounts have just this payment in them, leading to a heavy left skew of the

data. Also, the covariance of these features are also very high, as they are calculated from the same original data points.

This could be rectified by separating the data set into a set with payments other than the 250 and one with just the payments of 250. However, as this project is about clustering and unsupervised learning, this preprocessing step will not be taken. By design, the clustering should identify groups of customers, and as such by separating the data in this way will lead to an uncertainty in whether the clustering algorithm is identifying the trend, or if it is the human intervention. Further, this methodology is designed to be run on demand by non-data scientists, in this use case, the preprocessing explained here could not reasonably be conducted.

The average value for the gone-away category is 0.15610. As this is a binary, there is 15% of the data that is labelled as Gone-Away. This is a sign that the manual labelling of this is not sufficient, as the official estimates for the amount of CTF Gone-Away accounts are higher, 30%. It will be investigated whether reducing the number of non labelled gone-aways in the data set benefits the algorithm's performance.

The standout from the distribution of this is that most features are skewed pretty heavily to the left. Despite only 15% of entries being marked as Gone-Aways, it seems the majority of accounts are not paid into often, as the premiums frequency value is low, nor do the addresses get updated. This highlights the scale to which Gone-Aways are a problem, as it's likely from a cursory look that much more exist than are labelled.

3.5. Requirements

This project focused primarily using python code. When implementing please refer to the accompanying readme file for full library and read me list.

Python libraries used:

Pandas
SciKit Learn
Numpy
Geopandas
Matplotlib
Seaborn
MiniSOM

In addition, to calculate the traceability an API from t2a was used[3]. This is the person search API, and was used within Python, utilising Java to pull and read the resulting XML.

This project also used SQL to pull the required data from its source. This was using Opera.

3.6. Design and Implementation

In this section the design for this project will be walked through. For each part of the methodology, there will be design decisions made with justifications, then an explanation of the exact implementation used in this project. The methodology has been split into three areas, clustering, explainability, validation. Then the testing process will be explained. This methodology was designed with the dummy data set.

3.6.1. Clustering

The aim of clustering in this project is to create distinct customer sections based on behaviour features. These customer segments must be distinct, meaning they must be sufficiently different from one another, so as to not cause confusion by those that see the outputs. There must not be too many clusters produced, this is reflected in the literature where there are not many segments produced. The clustering should convey meaning, that being to tell information about the customers that is not obvious from initial analysis.

The algorithm should be able to distinguish clusters on large differences in a few features, rather than small differences on a lot of features. This is represented in the literature, where the customer segmentation is defined on a large difference of a couple of features. Further, the algorithm should have a way to automate the best parameter selection. This is so that the clustering can be run in the script without needing to set the parameters prior.

For all clustering the nature is unsupervised, as such the Gone-Away feature will be omitted, and will be used instead in the analysis of the clusters, the visualisations and the methods to label this feature better.

In the literature, the primary clustering technique used is K-Means. It is a simple algorithm, in concept and application, that is seen as powerful and

[3] https://t2a.io/products/people/person-search

effective. This approach is used for a variety of applications, utilising euclidean geometry theory to calculate the distance between data points and a determined number of centroids. The centroids position is at first randomised, as data is fed to the algorithm the location of the centroids is adjusted to fit with the assigned data points.

K-means is not suitable for categorical data. It is for this reason, certain data points are set to a value to represent the categories geometrically, this can be done in the preprocessing. For example, the sex feature has male and female set at opposing values, 0 and 1 and then another is placed in between at 0.5. For more complex categories K-means is not suggested to be used, but the algorithm can be adjusted, as suggested in the context section, to take account of this data type. This approach proposes to limit the use of categorical data, opting rather for quantitative features that describe behaviour on a more granular level.

In K-Means the primary parameter is the value of K, the number of centroids. The elected way to do this is to use the elbow method, an approach that produces a graph of the entropy (difference) between clusters. As mentioned, in the literature, this is particularly useful for customer segmentation as segments are required to be both as few as possible whilst still conveying meaning.

For this application, each centroid and the points it incompasses can be seen as a customer segment, so the features that most contribute to the centroid's location are those that distinguish the customer segments.

A second drawback of the K-means algorithm is that it is perhaps not complex enough to fully represent the intricacies of policyholder behaviour. In the literature, K-means performs well, but is clearly limited in some aspects.

Further, k-means struggles when it comes to explainability when using more than two features. This is because visualising the clusters produced by the algorithm requires the representation to be equal to the number of features. If dimensionality reduction is used, meaning is lost. Visuals will be further explored in the explainability section and in the results chapter.

This paper proposes the new use of an algorithm for the use of customer segmentation, the self-organising map (SOM). This is a clustering approach that uses the concepts of neural networks to reduce data onto a 2D plane of a set size. This feeds data onto a layer of neurons, associating with the most similar weightings, to find its location onto a map. The neuron's weight is adjusted to better fit the features, which here is the customer's behaviour, that it is assigned to. This approach gives similar data the same position on a map, and distances dissimilar data. To calculate the clusters the data points are assigned to, each unique neuron that is fired is assigned as a cluster, meaning the number of clusters determined cannot be set prior to clustering. This project will explore the use of K-Means to be used on the neurons within the SOM. This will reduce the number of clusters produced in the SOM algorithm.

The proposed benefit of using a SOM is that K-means has been an effective algorithm for customer segmentation as shown in the literature. However, much of the research identifies that its use is limited by the simple nature of the algorithm. SOM utilises similar ideas of similarity based clustering, but in application is very different, representing the data as a neural network. The use of neurons should allow for more information to be conveyed through the algorithm. Further, this project attempts to tackle the problem of explainability. For this purpose, the SOM algorithm produces visualisation of multidimensional data that is easier to understand than attempts of doing so with K-means, like dimensionality reduction.

This approach was performed using the Minisom package[4]. The parameters for this algorithm are the map size, (M x N), the number of iterations, the learning rate and the value of sigma. To determine the best combination, a grid search was performed, with the validation error and typographical error used. The validation error, the value of the difference between the data and the assigned neuron, is to be minimised and is the primary measure but the topographic error, the distance of a data point to the winning neuron compared to the second best, will also be considered.

[4] https://github.com/JustGlowing/minisom

The map size must be of suitable size so to be understandable, so will be limited to 50x50, this is admittedly arbitrary, but in the visualsation of the SOM it can be seen that this size diminishes returns. Further, the map will be kept to a square size. This is again to keep the visualisation simple, but is also to reduce the computation requirements of the gridsearch.

To visualise the SOM, the grid will be coloured to represent the weightings of the neurons and the data assigned to that square. This will return a topological looking grid that conveys the similarity of each square. This can be seen as a validation for the difference of the clusters produced. Also, a visual that overlays a pie chart for each square with the percentage of Gone-Aways labelled will be produced. This will give an initial understanding of how the neurons reflect the Gone-Away information. Finally, the SOM will be coloured with the clusters, to give an understanding of how the SOM has distinguished groups.

A downside to the SOM is that it is subject to identifying the outliers in the data, meaning the remaining data is grouped together in a few very large clusters. For this, it has been found effective for SOM to be run again individually on the areas that contain large proportions of data. Effectively this is an approach of progressive clustering. This will map out one area into its own map to identify the differences within. The K-means algorithm will rectify any smaller clusters.

3.6.2. Explainability

Explainability is crucial to this project and must be considered as a part of the validation process, as such it performed directly after the clustering. For the following step of validating it is important to not only validate the clustering but also validate the reasoning. The outcome expected by Ai-London is a reusable approach that produces explainable customer segments. Explainability here will be distinguished into two similar but crucially different meanings. The first being to explain why a customer was assigned to the segment, this is beneficial to understanding the customer base within a segment, and in the research is identified as interpretability. The second being explain what makes a segment meaningful to who is looking at it. This is what policy within the insurance company will be based based upon. Separating

these out is not mentioned in the literature explicitly, but can be extracted from how it is proposed the meanings are used.

Currently in the literature, this is primarily a method of subjectively choosing explanations based on the features in the algorithms. This is not deemed as suitable for this approach, as the intention is to automate, and a range of more concrete methods will be proposed here.

Two methods are tested in this project, both being model-agnostic approaches. The first being a simple ad-hoc statistical test, that will compare features of a cluster to determine which are outliers and therefore important. This is based on the decision process explained in the customer segmentation literature. The second will be a take on the current state of explainability in unsupervised learning, the use of a decision tree to determine a hierarchical process.

The first proposed approach is a statistical one. The test used will examine whether the sample, the points assigned to a cluster, are representative of the whole dataset for a particular feature. If it is not representative sufficiently, it can be assumed that the feature is statistically important in defining the cluster. The size of the clusters are expected to be large, which can affect the results, wrongly assigning importance for those clusters which are of smaller size. Also, this approach may not be complex enough to fully understand the intricacies of the clustering used, the decision process will not be explained by this approach, opting to just give a reason as to why the customer has been assigned to that cluster.

The statistical test will be similar to a normal distribution Z-test. The mean of the features within a cluster can be identified as the sample set, and will be compared to the whole dataset, the parent population. If it is a standard deviation away from the mean or more, it is reasonable to assume that the value is statistically important and therefore can be said to be important to that cluster. In this, it can also be deduced if the value is relatively high or low, giving more insight to the end user.

The second approach is to use a classification machine learning model that can be trained to predict the cluster that is assigned. This will be used in the validation section of this project as well. The algorithm selected should be one that is designed to convey the reasoning behind its decision. The reason can

be easily extracted and the feature importance according to that algorithm can be interpreted as the reasoning behind the clustering.

The literature suggests using a decision tree for this process. This will allow for both the deduction of important features within the cluster, but also for the reasoning of the decision process to be seen. It will be hierarchical, so it can identify which features are the most important for each cluster.

3.6.3. Validating

Validation in this report refers to two separate processes that are performed after the clustering. The first being the validation of the clustering itself. It is assumed that at this point the algorithm's parameters have been optimised to fit the data set. It also refers to the validation of the explainability. This can also be seen as an output of this methodology, and will be represented primarily by visualisations, to be interpreted by the end users.

To validate the clustering itself, a separate machine learning model should be used. This is one approach that is suggested in the literature, and is deemed the most applicable in this project as the approach should be automated. The model will take the same data as the clustering and predict which segment the clustering algorithm has been assigned to. The prediction score can be computed as the accuracy, and it can be determined that the clustering is valid if this score is high. To understand why, the classification model can be seen as using a different approach to try and make the same decisions. If this is possible, the clustering algorithm has identified real patterns in the data. The data is assumed to already be preprocessed for the clustering, so no further work is required for the ML algorithm to predict.

A decision tree is proposed for this purpose. This algorithm is ideal for this project for many reasons. The decision tree uses easy to understand decision making, allowing it to be used for the explainability section. The decision tree can also be visualised, which allows for the end user to get a better understanding of how a cluster was decided upon.

In visualising the decision tree, certain things should be kept in mind. The goal of this visualisation is to determine which features are for which customer segment, and such what are the defining features of that segment.

The decision tree will be limited in size, as the visualisation must be understandable, but will likely reduce the accuracy of the prediction. It will be important to examine the hierarchy of decisions, as this will indicate which segments are closer together.

To further prove the clustering is valid, and to make this understandable to the end user, visualisations should be produced. The goal of these is to explain why a certain feature is unique to that cluster. These should represent the distributions of the clusters, and indicate how they differ. The construction of these visuals will be based upon the feature being analysed, and will stick to the important features identified in the explainability part of this project. This will then be used to validate the selected method of explaining the clusters.

The final approach to validating the clusters will be to examine how well it reflects the gone-away problem. The logic of this being that gone-aways are a known problem in the data, but the scale of which is unknown. The methodology here should identify underlying patterns in the data and these patterns should be reflected in the determination of a gone-away account. In this project, the likelihood of a cluster containing a gone-away account will be assessed, looking at the number of labelled gone-aways within that account as well as looking at the importance of known gone-away behaviours to that cluster assignment.

3.6.4. Testing Process

The testing process will begin section by section as the automated script is created. Each section of the methodology has its own testing process that has already been highlighted. The clustering algorithms will use their similarity metrics to ensure the clusters are sufficiently different. The explainability methods will be compared with one another using the validation methods explained. This will all be performed on the dummy data set, where it is possible to manipulate the data to test different scenarios.

Once this is complete, a script will have been developed that encompasses this entire methodology. This will allow for the approaches to be tested on the real data set. The same metrics as in the dummy dataset will be examined. The results should convey real meaning, allowing for confidence in the methodology, as well as highlighting areas that will need to be developed

further. This will be tested by examining the outputs and how well they characterise the problem of Gone-Aways.

The computation required will be taken as clock time. Whilst this project is designed to be run on an on-demand basis, so the computation time is not very important, this is still designed for a business use case where cost does matter. Computation resources used is not appropriate as when this project is implemented as a product, the product will be run on a server, and optimised to do so.

3.7. Example Implementation of Methodology

As this project is proposing a full process of clustering and analysis, it is worth explaining the full process. This is also the exact process the script will follow, the script will be modular, as to allow the use of different clustering algorithms and explainability approaches to be integrated.

The data being inputted into the methodology should contain data points that explain behaviour, and should be categorical. The dataset should be the full body of data, but can be adjusted to be better represented by distance based algorithms. The first step will be to perform the clustering. In this project K-means or SOM is used. The clustering approaches parameters will be automatically selected, through the use of a grid search or another algorithm to determine the minimal error score. The clustering will be validated by a classification model, where the accuracy should be maximised.

The clustering will then be made to be explainable. Here this means using a model agnostic approach. This approach should determine what makes the clusters unique, and should be understandable by a non data scientist. Visualisations throughout the process should be produced. These visuals should be sufficient for a third party to understand the clustering results, clearly describing distribution of clustering. The explanations and visuals should help to create a labelled data based on semi-supervised principals.

4. Results

4.1. Analysis of Outputs

The results displayed in this section will focus on the dummy data set. In this section, the focus is on how well the methodology solves the task of clustering and customer segmentation. The performance of the clustering will be examined initially from a machine learning perspective, examining how effective the clustering is at distinguishing distinct groups within the data. Then the explainable cluster approaches will be explored, identifying the benefits and limits of each. The validation techniques will be examined from the lens of how well the approach has validated the clustering as well as how well the clusters have been explained.

In this section, the dummy data set is used. This allowed for the benefit of manipulating the data to best illustrate the results being described. This being said, unless stated otherwise the results are based on the default values explained in the data section of this report. However, using the dummy data makes validation difficult. For this reason, the validation and labelling Gone-Away aspects will be examined once the real data sets are utilised.

The clustering techniques themselves are not novel, but their application in this domain is. The use of explainable clustering on these algorithms is not new, but the use of these approaches for customer segmentation is. The development of a script combining these approaches is a new development as far as public information is available, but could exist within private organisations.

4.1.1. Clustering

In the methodology, it was explained that the clustering technique used must be versatile, it must provide adequate difference between the clusters whilst not producing so many as to convey too granular of information. Two approaches were explored as a part of this project, K-means and SOM and then a hybrid of the two. It was found that the hybrid approach fitted the criteria of this project the best, but K-means with alterations performed well.

When k-means was run on the data set, it was discovered that the elbow method through the use of Kneedle worked well at identifying the best number of clusters. However, this approach did return some results that provided more clusters than the literature deemed appropriate for customer segmentation. In Figure 1, the elbow plot for the data set can be seen, in this plot, kneedle identified 12 as the optimum number of clusters, for use in customer segmentation this was limited to just 10, which had a similar inertia score. This highlights an issue with using the kneedle algorithm, as it can be seen at k=12 there is the point before there is a bump in an otherwise smooth curve. Using a geometric approach, this bump was identified as a turning point, in turn labelling it as the kneedle point. In this case the distance scores between the numbers of clusters 10 and 12 is negligible. With 10 clusters on the dummy data, the maximum cluster size was 1618, 16% of the data, and the minimum size was 600, 5% of the data, indicating that no cluster was too large or too small.

In the tests, the k-means approach is computationally efficient, the whole k-means part of the script running in 26 seconds, which is acceptable for this application. In figure 2 a TSNE conversion of the input data plotted as a scatter plot can be seen. TSNE is appropriate dimensionality reduction here as it samples based on distribution, allowing for the patterns in the data to be preserved. The effect is better seen in 3D, but the K-means algorithm did well at separating data into distinct groups. When looking at the figure, the group of clusters 0 and 4 in the top right is in distinct planes on the z-axis.

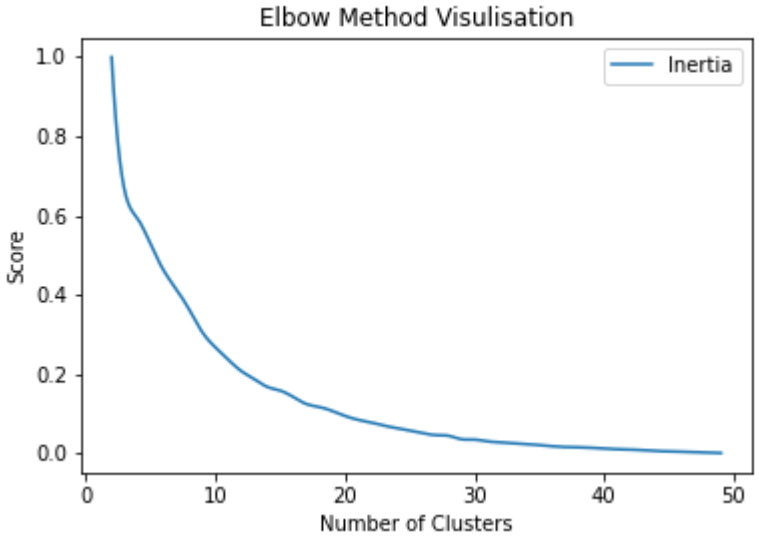

Figure 1. Elbow Method for K-means

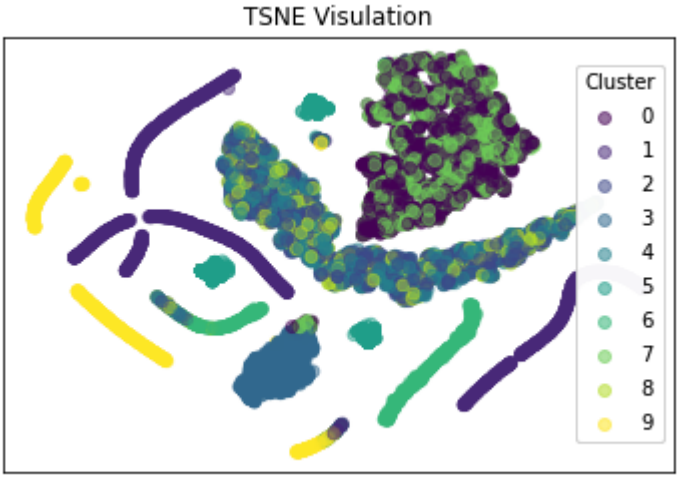

Figure 2. TSNE visual of K-means

When analysing the Self-Organising Map algorithm it is important to note that this is an approach that reduces data onto a 2D plan, and is designed to map similar data points close together, and dissimilar points far. To understand how the algorithm has performed it is crucial to examine the created distances between data points.

The best way to run the SOM for this application was to perform a gridsearch. The quantization error was determined to be the best measure of performance, as just measuring the topographic error, the distance between neurons,

provided similar results for a wide range of parameters. It was identified that through adjusting the grid size, the number of clusters could also be manipulated, so to fit the needs of this project this parameter was limited to a maximum of 50x50. Sigma and learning rate were searched between values of .5 and 5, finding that these values varied as the number of iterations was adjusted. The number of iterations refers to how many times the data was looped through the SOM algorithm for it to model the data better. It was found that after 1000 iterations there were diminishing returns.

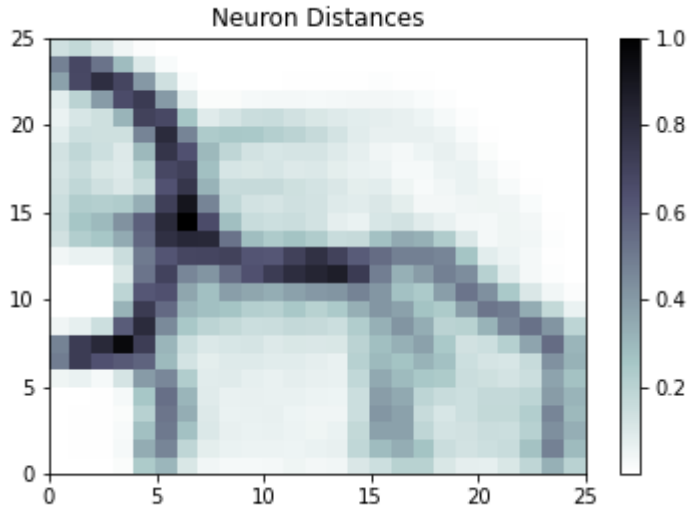

Figure 3. SOM neuron distance plot

Figure 3 shows the distance map for the SOM run on the dummy data. To interpret this figure, it is worth imaging as a diagram of a hill. The darker the area on the grid, the steeper the hill. So, it can be interpreted that points with coordinates (10,5) and (20,1) are relatively close, but (10,5) and (10,13) are relatively distant. This gives an understanding of how the SOM has organised its neurons.

Through manipulating the dummy data distributions, it was quickly identified that clustering purely with the SOM provided a large number of clusters, when clustering was performed purely on the winning neurons. In figure 3 it can be interpreted that every square is its own cluster. Many contained a small amount of data (1%-) and then a couple contained a large amount (50%+) This can be interpreted as small groups of very similar points. Performing a

second SOM on these larger clusters allowed for more manageable clusters on the larger groups, but just added to the number of clusters. To remedy this, it was deemed necessary that for the application of customer segmentation, an additional algorithm was required to be performed on the neurons, so a hybrid approach using K-means was used. The application here was similar to the pure K-means approach. The results of this can be seen in Figure 4.

In Figure 4, the neurons that have won a data point have been colour coded to which cluster they belong to. It can be seen that the hybrid has performed well, providing distinct groups, and solved the problem of too many clusters being produced. There is a larger range of cluster size in this approach, with a range of 500(5%) - 3000(30%), which could be an issue in producing distinct groups of data points, and will be explored in the validation section. This visual of the SOM is much clearer to understand than the dimensionality reduction of the K-means algorithm.

Overall, with the K-means reduction, the algorithm typically took 75 seconds to run, indicating that this is more computationally costly. However, this is largely in the gridsearch, and such could be optimised by running this in parallel, which will greatly reduce the wall clock time, at a cost of extra resources needed.

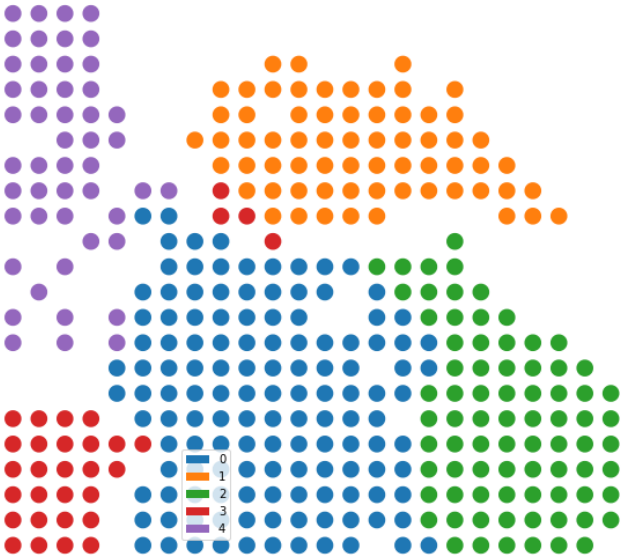

Figure 4. SOM clusters plot

Overall, from these initial tests, the basic SOM can be deemed as not useful in the application of customer segmentation. From this point, both K-means and SOM with K-means will be explored further in the validation stage, with a preference for the SOM with K-means. This can also be seen as exploring whether a larger range of cluster sizes is detrimental to explaining the clustering. In a more general sense, the distance based algorithms performed well at defining distinct groups of similar features.

4.1.2. Explainable Clusters

For this project, three distinct methods are used to explain what segment of the customer the clustering algorithm has used. The techniques used will be analysed here through quantitative and visual techniques, with the goal of

understanding which approach is most applicable for the automated approach for this project.

From the literature, the current primary method to explain unsupervised learning is to layer a supervised technique on top. As there is a wealth of research into explainable AI for supervised learning, it follows that by using a classifier the same techniques can be used. This project used a decision tree to predict from the input data which cluster it will be assigned to. The decision tree for the dummy data performed using SOM can be seen in Figure 5, this particular tree was visualised using dtreeviz, trained using k-fold and with an accuracy of 99.8%. From this figure, it can be understood a process that can be followed to come to each decision. A histogram at each point for the classification determines at which point a distinction was drawn between each cluster. For example, it can be understood that for the decision to classify into cluster 1, the duration between payments, maximum premium payment and duration between moving must all be low for those decision groups.

Figure 5. Decision Tree of SOM

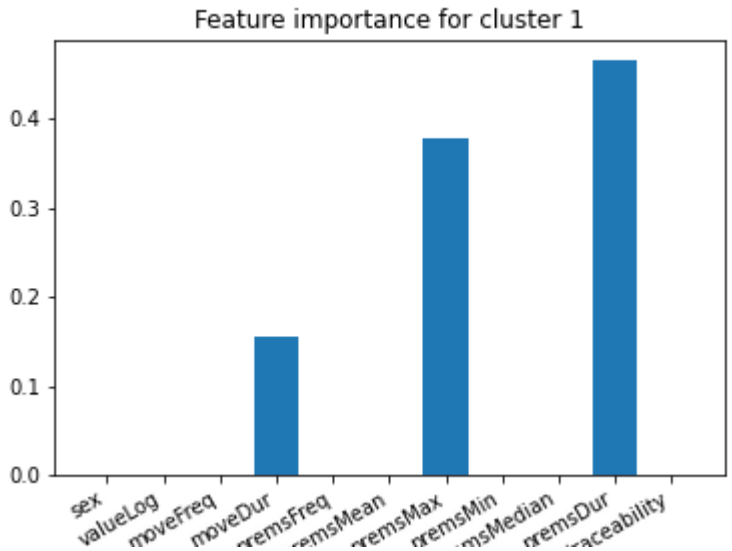

Figure 6. Feature importance bar graphs for cluster 1 of SOM

To understand this process further, the importance of each decision can be tracked, and then plotted onto a bar graph, like in Figure 6. This highlights the relative importance in making a decision, so it can be seen that the duration between premium payments is the most important feature in making this decision, but the other two features are also important.

By limiting the decision tree, following the literature, the explainability is improved, but the predictive ability is improved. This can be seen in Figure 7 and 8, in which the decision tree was trained on the K-means clustering. Depth was set to 9, the path is much harder to follow, and the difference between feature importance in Figure 8 for cluster 8. Too many features make up a decision in a tree this large to be explainable, which follows the current literature.

This also highlights the issue of K-means producing a larger number of clusters. With more clusters, more reasons are needed to explain why the clusters are different. For example, a tree of depth 4 is the maximum set in this implementation, which has the maximum of 32 branches. Assuming at least three decisions are needed to separate clusters, a maximum of 10 segments can be identified. With a tree of depth 9 like in this example, there is a maximum of 1024 branches, showing how incrombehensibe this can become.

For the K-means algorithm, the explanations are very simple, focusing on just a couple of points. This reflects that the K-means approach is not very complex, that purely uses the average distance to a centre point. This did not identify the more complex patterns that an algorithm like SOM did.

Another drawback is whilst this might be true for how the decision tree has defined the clusters, it may not be representative of real differences between clusters further down the tree. Examining figure 5, one of the most predominant features is move duration. If the reason that all features are different is the same, little is learnt. There is too much nuance to be comprehended easily, a different reason for each cluster is preferable.

Figure 7. Decision tree visual if depth is high

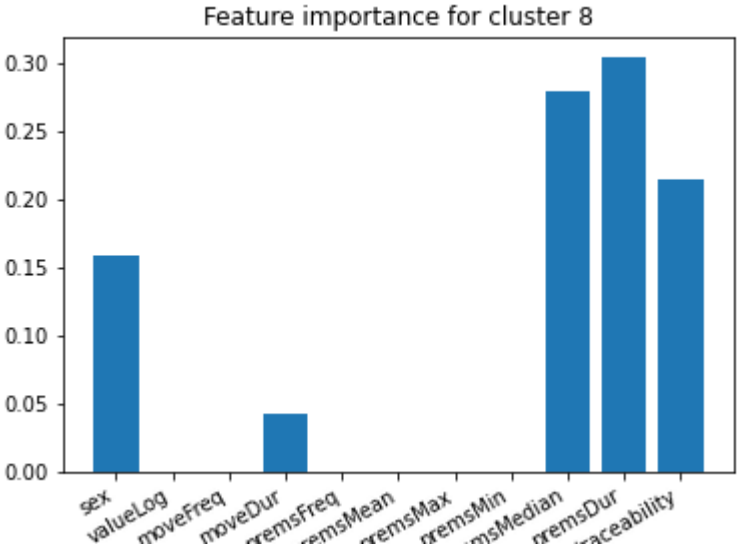

Figure 8. Feature Importance plot for Cluster 8 of K-means

Another way to explore the difference between clusters, and therefore produce an explainable reason for the clustering is to use a statistical method. By examining the feature makeup of each distinct cluster, it can be observed where the differences are. Figure 9 highlights this, by plotting the overall distribution of a feature and then showing where the mean values are, the differences can be observed.

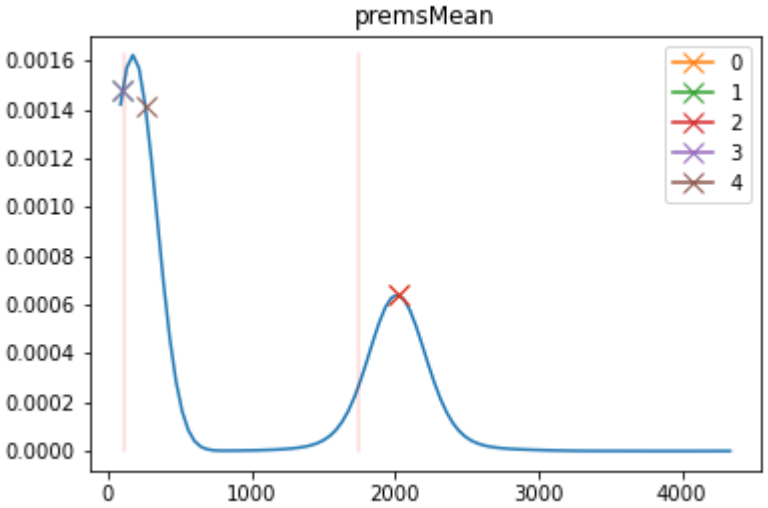

Figure 9. Distribution plot of mean premium payment for SOM

Figure 9 shows the distribution for the mean of premium payments, and from this, it can be determined that cluster 2 has a high value for this. It was a design choice for this methodology to accept non-normal distributions in the data, so the two peaks of this feature can be clearly seen. In this, the majority of clusters align with the first peak with the majority of the data, so this is not an important feature for this cluster.

To calculate this, the standard deviation can be plotted, which are the red lines in the plot. Anything outside of those marks are outliers. Figure 10 has mapped a feature onto a normal distribution to illustrate this point further. Cluster 4 has a distinct small amount for the value of an account. By visualising in this way, it can be seen how this is comparable to a z-test in statistics, where the null hypothesis to be tested is feature x is not important to cluster y, to disprove this hypothesis, the point must be outside of the red boundaries.

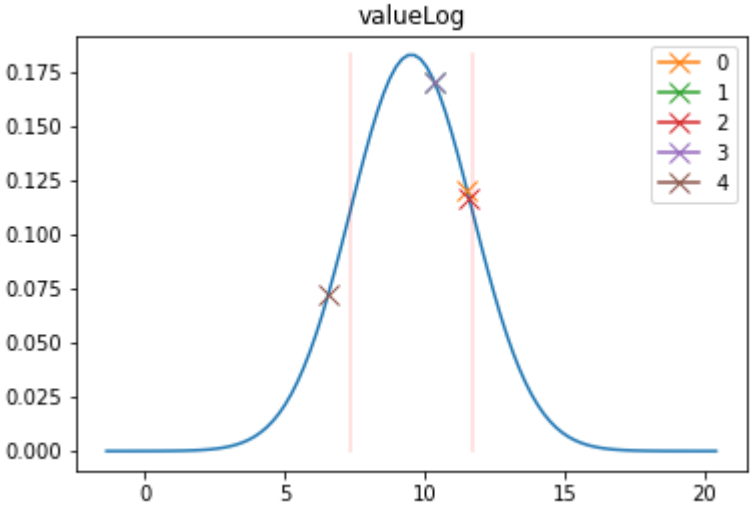

Figure 10. Normal distribution plot of the policy value for SOM

The importance according to this approach for the SOM clustering is as follows:

Cluster 0: high premsMean, high premsMax, high premsMedian.

Cluster 1: high moveFreq, low moveDur, high premsFreq

Cluster 2: high moveFreq, low moveDur, high premsMean, high premsMax, high premsMedian

Cluster 3: high premsFreq

Cluster 4: low valueLog, high premsDur

This approach provides an easy to understand way to understand the make up of clusters in general. For example, it can be understood that cluster 3 contains customers who are likely to pay into accounts often, and average every other value. To contrast, cluster 0 has an average account value, but is likely to pay large sums of money into their accounts. This method can be seen as a statistical way to describe the radar plot in Figure 11, which plots all features and all mean values by cluster together. This radar plot is similar to current industry techniques of displaying customer segments.

The key downside to this approach is that some meaning can be lost in the statistics. For example, in Figure 9 it can be seen that because the values in cluster 4 have weighed down the overall average of the valueLog feature.

Because of this, no cluster can be assigned to contain high policy value customers. This is something the decision tree approach did pick up, as it allows for less important features to still be valid, using it to separate data into Cluster 0.

Further, this approach requires a large amount of data to be sufficient. Statistically speaking, distributions can only be modelled using normal distribution if they are sufficiently large enough for the central limit theorem, the theory that the sum of probabilities create a normal distribution, to apply. With a skewed distribution, this is necessary to apply for this approach.

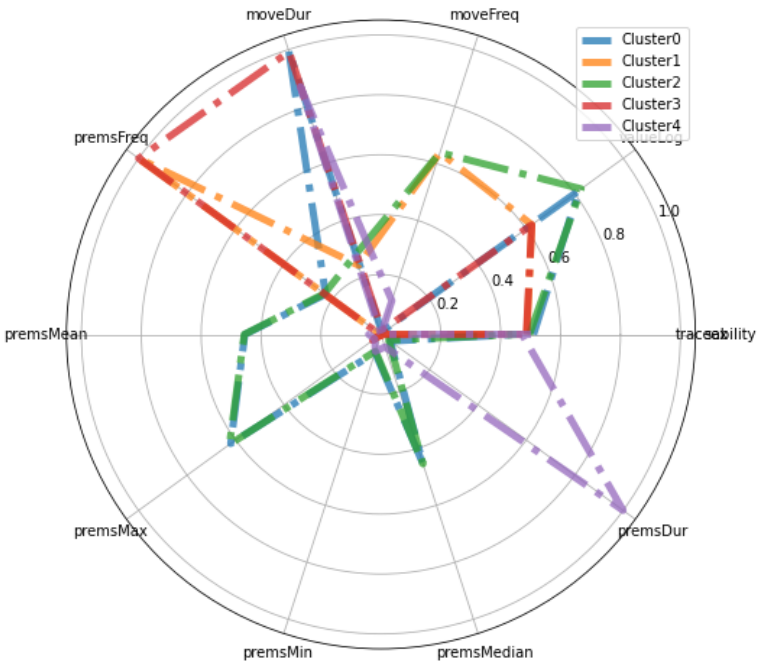

Figure 11. Radar plot of SOM clusters

4.2. Other Datasets

The previous section shows the development of a working approach for customer segmentation that has been compiled into a script. This script can be seen in the appendix, it contains the option of K-means or the SOM clustering algorithm, requires

the data to be submitted as a dataframe and allows for the identification of what the features to be labelled as. The default setting is to use SOM, and the statistical explanations as this has performed best in the tests. The code outputs visuals that can be seen in the prior section and this section, as well as a text explaining the important features for each cluster.

In this section, the script will be run on the real dataset, after minimal preprocessing as explained in the data section, and the results will be analysed here.

Using the SOM algorithm, there were 10 clusters produced. These can be seen mapped in Figure 12. From this figure, it can be seen that the clusters are distinct. Comparing Figure 12 to Figure 13, it can already be identified that the clusters most likely to contain Gone-Aways using this approach is cluster 5, with cluster 1 and cluster 9 also likely to contain Gone-Aways. Using this approach, the clusters are of a wide range of sizes, similar to using the dummy data, the largest cluster being number 2 containing 42% of the data, and the smallest containing 2% of the data.

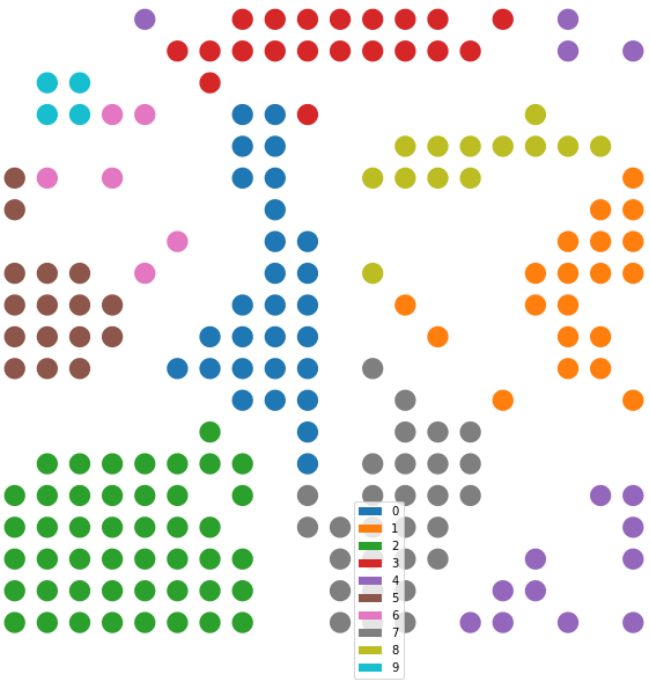

Figure 12. SOM for real dataset

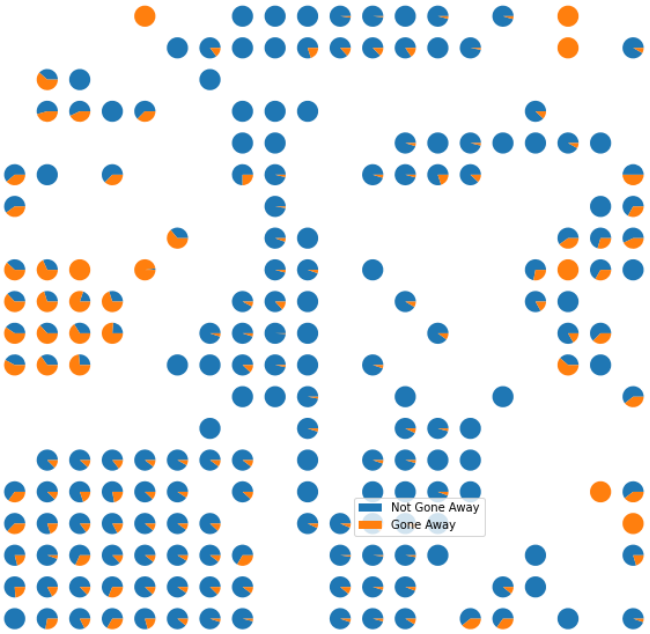

Figure 13. SOM showing proportions of Gone-Aways

```
Statistically
Cluster 0 has relatively low feature sex with mean value of 0.886814232121672
Cluster 0 has relatively high feature premsFreq with mean value of 0.001981234566319549
Cluster 1 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 3 has relatively low feature sex with mean value of 0.886814232121672
Cluster 3 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 4 has relatively high feature moveFreq with mean value of 0.0036713230657418
Cluster 4 has relatively low feature moveDur with mean value of 0.9928232096066413
Cluster 5 has relatively low feature traceability with mean value of 0.8828596774365935
Cluster 6 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 6 has relatively low feature traceability with mean value of 0.8828596774365935
Cluster 7 has relatively low feature sex with mean value of 0.886814232121672
Cluster 7 has relatively high feature premsFreq with mean value of 0.001981234566319549
Cluster 8 has relatively low feature sex with mean value of 0.886814232121672
Cluster 8 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 9 has relatively low feature sex with mean value of 0.886814232121672
Cluster 9 has relatively low feature traceability with mean value of 0.8828596774365935
```

Figure 14. Statistical eplanations

When looking at the explainability of this approach, the outputs of the statistical approach show that the most important feature for cluster 5 and 9 is the traceability score, which is relatively low. This score tracks with the knowledge that Gone-Away accounts are by necessity hard to trace. The distinguishing feature for cluster 1 is that it has a high duration between premium payments, which is a noted feature of Gone-Away accounts.

Looking at the inverse, it is interesting that cluster 0, which is characterised by high accounts and plenty of premium payments, when looking at the SOM, contains very few Gone-Away accounts.

When analysing the decision tree approach, seen in Figure 15, to explainable clustering, the validation score for this clustering approach had a score of 99.9%. This indicates that the SOM clustering is valid. When following the decision tree, the first decision is telling for Gone-Away accounts, it distinguishes between the accounts that pay into an account often and the accounts that don't. Further, using this approach, cluster 6 can be identified as likely Gone-Aways, as they are relatively low for the number of times their address has been updated. This clears some ambiguity in the SOM visual, where it is unevenly split between Gone-Away or not for this cluster. Interestingly, cluster 4 is the most often occurring result when analysing the final decision along the branches. This cluster is characterised by containing individuals that move frequently and update their account accordingly. So this cluster is unlikely to contain Gone-Aways.

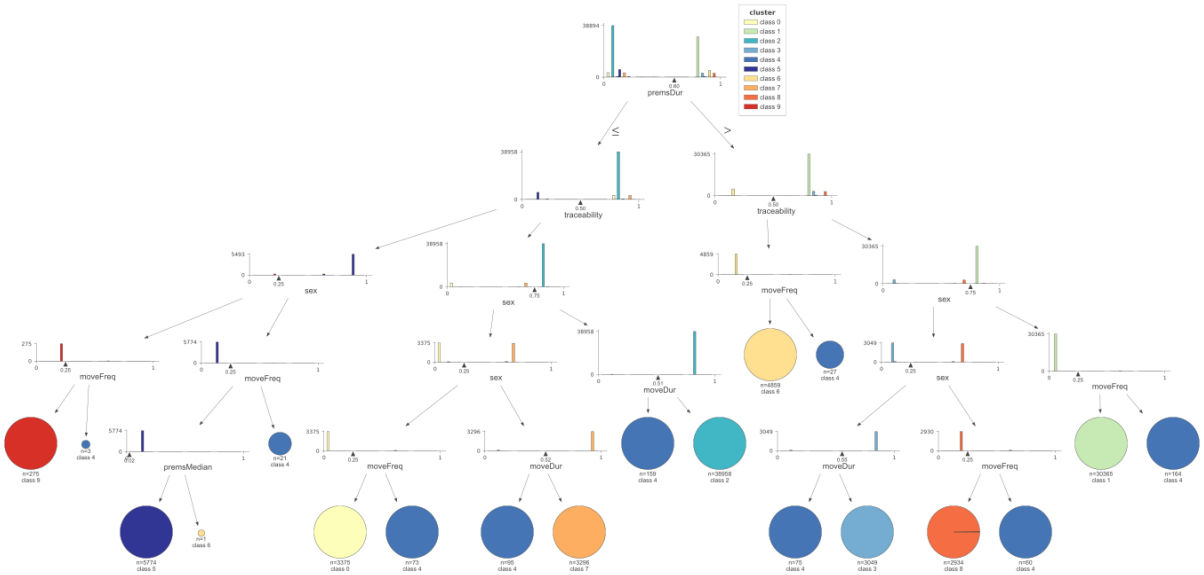

Figure 15. Decision tree of real dataset

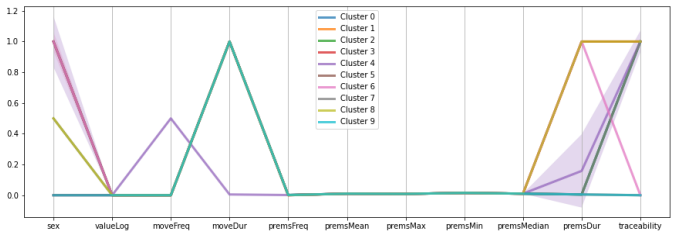

Figure 16. Parallel plot of real dataset

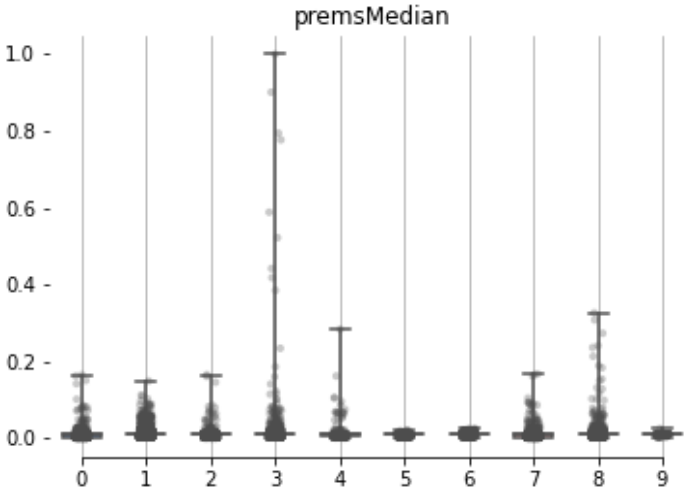

Figure 17. Boxplot for median premium payments on real dataset, showing clusters

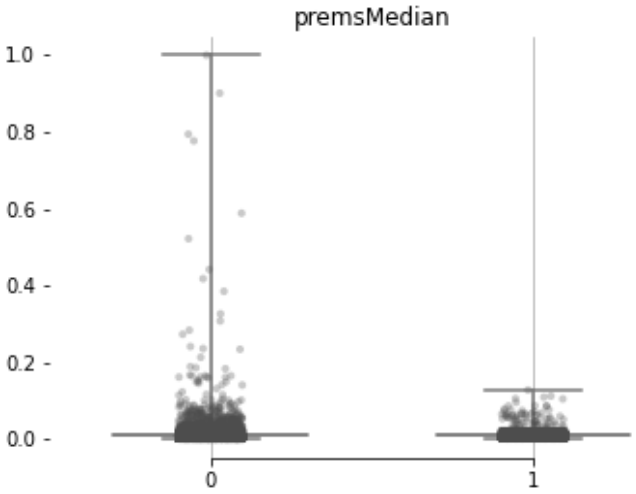

Figure 18. Boxlpot for median premium payments, showing distribution of Gone-Away accounts

The decision to do minimal preprocessing did affect results of the clustering. This can be seen in the parallel plot in Figure 16, where the heavy left skew of premium payment amounts, caused the features between clusters to be indistinguishable.

Drilling down into these features further, Figure 17 shows a boxplot for the median premium payments. The dots indicate individual data points, and it can be seen that a few points that lie within cluster 3 skew the whole distribution. Comparing this to Figure 18 which shows the same for labelled Gone-Away data, the majority of clusters more resemble the Gone-Away class.

For analysis, the K-means algorithm was also tested on the real data set. Using this approach 6 clusters were identified. The elbow plot in Figure 19 shows that this is a very distinct turning point at which the inertia stagnates for a few cluster sizes. When looking at the explainability approaches, they both produced similar outputs, with simple decisions. This reinforces the point that K-Means is too simple an algorithm to distinguish between complex patterns within the data. From the decision tree in Figure 21, it is clear that the clustering was performed primarily on premium payment frequency and traceability score only.

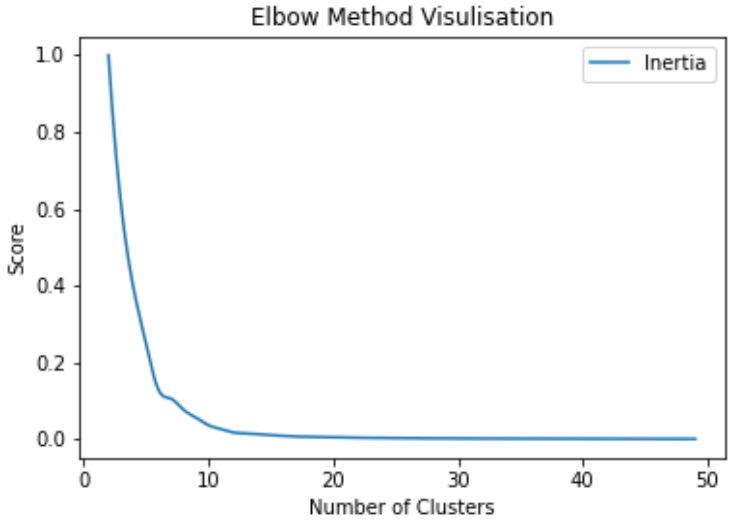

Figure 19. Elbow method visual fro motor insurance dataset

```
Statistically
Cluster 1 has relatively low feature sex with mean value of 0.886814232121672
Cluster 1 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 2 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 3 has relatively low feature sex with mean value of 0.886814232121672
Cluster 3 has relatively high feature premsFreq with mean value of 0.001981234566319549
Cluster 4 has relatively low feature traceability with mean value of 0.8828596774365935
Cluster 5 has relatively high feature premsDur with mean value of 0.45131866073000315
Cluster 5 has relatively low feature traceability with mean value of 0.8828596774365935
```

Figure 20. Statistical explanation of motor insurance clustering

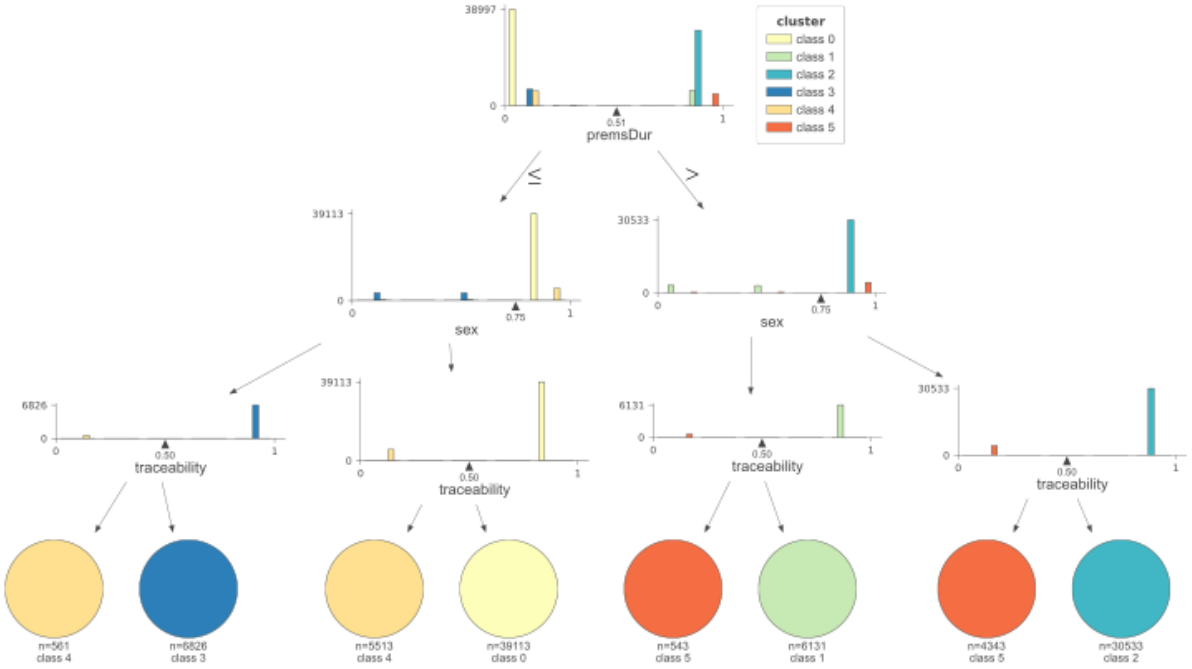

Figure 21. Decision tree explanation of motor insurance dataset

Finally for this analysis, a third dataset will be tested. This is a motor insurance set, and this analysis will aim to determine the characteristics of a fraud claim. For this, the SOM algorithm was used. The dataset is significantly smaller, so it was necessary to add a number of iterations to the gridsearch of SOM parameters, allowing for better training on this data. The SOM coloured with clusters for this dataset can be seen in Figure 22, there were 8 clusters found with this dataset. The SOM with this dataset is less clear, and the prediction score of the decision tree trained was also lower, at 88%. This is likely because of the dataset size, suggesting that datasets used with this approach should be larger in size.

Figure 22. Display of clusters from SOM algorithm

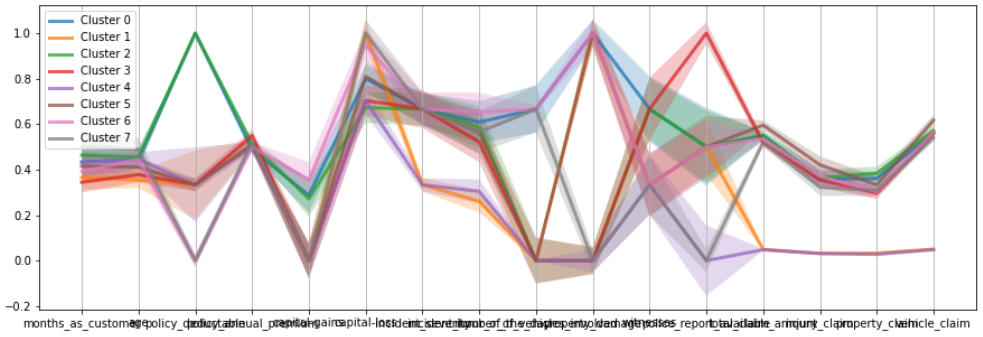

Figure 23. Parallel plot of SOM clusters on motor insurance dataset

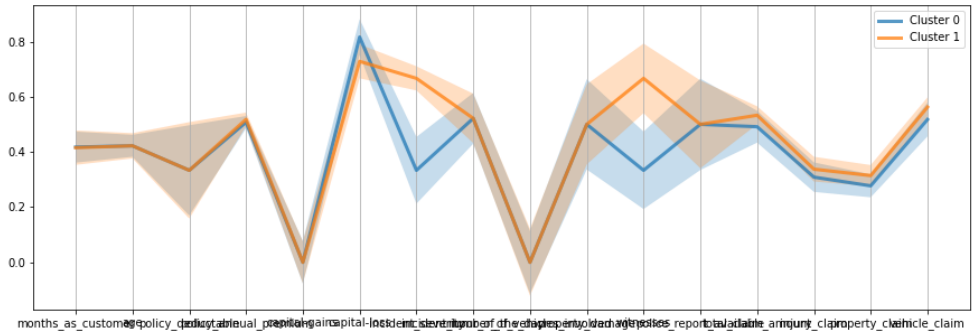

Figure 24. Parallel plot of fraud against not fraud cases

For this task, the desired outcome is an understanding of understanding motor insurance fraud customers. Fom Figure 23 it can be seen that fraud and non fraud classes are very similar, with non fraud cases being likely to have slightly smaller values for incident severity and claim damage, with a larger value for police report available. The shadow of the plots show the decile for that class, and throughout the plot the shadows are broad, indicating that there is a large amount of variance within the classes. This combination of results means that it is extremely difficult to identify differences between the classes based on these features.

When looking at the clusters in Figure 22, it is clear that most clusters are pretty similar. The lines mostly follow the same pattern, and have large shadow plots, indicating that the clustering algorithm has not made a distinction between these features. It appears that clusters 0 and 5 loosely follow the trends that the fraud class does, and cluster 7 seems to follow the shadow of the fraud cases. When looking at the statistical explanation, cluster 5 did not return any explanation, meaning it follows the trends of the dataset as a whole, and cluster 0s explanation was that the deductible of the policy was relatively high. This indicates that this explainability approach cannot achieve the granularity needed for a complex problem like this.

The key takeaway from this dataset is that the clustering algorithms used are not sophisticated enough to produce meaningful results from a small dataset. The low validation score of predicting the clusters indicates this. When the clustering algorithm performs poorly, the explainable methods are negatively impacted, it is hard to explain something that is not performing. The problem here is the dataset size not being compatible with the clustering approach, which has led to poor explainability. This conclusion can be made, because the prior dataset tested in terms of distributions

of the features and variance between features was worse, indicating that the clustering algorithms can perform well if there is sufficient data to learn from. Despite this, the script created has allowed for some clustering to be done. The approaches should be tested on a larger dataset to achieve stronger results.

4.3. Evaluation of Results

To conclude this section of the report, each section of the methodology will be summarised, with a focus on how each part leads into the other.

To begin, the algorithm outperformed K-means for the problem of customer segmentation. The SOM algorithm achieved a deeper understanding of the data, revealing hidden patterns in both the real and dummy datasets. The SOM algorithm also allowed for easier visualisations to aid understanding, and a deeper understanding of the labelling of Gone-Away problem. Both algorithms were able to be validated and predicted using a classification model.

The explainability approaches used in this report both performed well. Interestingly, in almost every case, except the K-means on the real data, they provided different explanations for the clustering. This indicates that these approaches used sufficiently different methods of modelling differences in the cluster's features. The statistical approach provides a more out of the box easy to understand explanation, whereas the decision tree requires slightly more analysis to understand. However, the decision tree provides more granular reasoning showing the exact differences between clusters. A combination of the two could be utilised, as they did not produce contradictory explanations, that allows for a surface level analysis and a deeper understanding, which would allow for the end user to choose which is preferable.

The automated approach worked well. The script produced which was run on the real data allowed for easy clustering and sufficient explanations. The visuals produced could be improved in some ways, like making the colouring of the clusters uniform across graphics, but work to aid analysis and understanding of the decisions the machine learning models made. This script, with some alterations provides a good proof of concept for Ai-London to create a product from.

5. Discussion

5.1. Research Questions

For this project, it was necessary for the research questions to be more broad than is preferable for a project of this kind. This was because the research within this project had a wide scope, encompassing many different research areas that are each undeveloped in the literature, being clustering for customer segmentation and explainable unsupervised learning. The first research question, of utilising clustering for customer segmentation for insurance data has been realised. Both clustering techniques defined distinct groups of customers with clear explainable behaviour, and even indicated how to better label the problem of Gone-Aways, providing evidence for the second question.

The problem of explainable clustering has not been fully realised within this research. The approaches used in this project can be combined to give a good understanding of the clustering and explain the distinct features of the groups of customers, but there is no evidence that the explanations given will be useful for the purpose of this project, to inform policy. This will be improved in future work with Ai-London, when a client receives the work and feedback can be given. The outcomes of this project did model some existing industry work in customer segmentation, but as this is a new area for the industry, there were no set practises to compare or model.

The script produced to make the clustering and explanations works and provides a good proof of concept that it is possible that this process can be automated. For this to be turned into a full product, it is necessary that this is refined, optimised and the visuals tailored to the clients needs. The script is designed to be transferable for different datasets, and is modular, as to allow for different approaches to be developed and integrated. The work in this project is a good proof of concept that the methodology in this project will achieve the desired outcome for most datasets, and also provides insight into what kinds of datasets won't be effective, namely datasets of small size.

5.2. New Knowledge

This research has provided an insight into how clustering is effective with financial/insurance data, and more generally data that describes an individuals behaviour. For this, clustering is an effective approach that does distinguish distinct groups of customers, but fails to describe why these clusters exist. For this technique to be adopted by industry, it is necessary to produce approaches that explain the clustering processes, aimed for non-datascientits to interpret. In this, this research has made the first steps.

Using post modelling explainability approaches are effective, this allows for the flexibility of different clustering techniques to be used, but also allows for explanations that are not obvious from the machine learning model used, even if the model can be deemed as transparent. This is evident in examining the likelihood of Gone-Away accounts that were not identified when just examining the SOM, and also in how the K-means algorithm could be explained better than using something like dimensionality reduction. The decision tree approach works well at explaining a decision being made, but can be more complicated and confusing for a non-datascietist to understand. The statistical approaches are easy to understand, but can be underdeveloped in actually providing reasonable explanations. Ultimately this research has reflected that there is a trade-off between simplicity and comprehension that neither approach in this research achieved to bridge.

It is known that visualisations are required to aid in explaining clustering. These visualisations allow for an understanding of the distributions within clusters, and allow for human interpretability. This research has shown that it is not yet possible for a fully automated approach to be used, and visuals produced in this research have provided a tool that can be used by a human end user to interpret the clustering as well as the explanations. Visuals that examine distributions of features to be compared within clusters are effective, and provide explanations for clustering that can be understood quickly.

Further research is required in the development of explainable clustering techniques. The approach developed in this project is transferable, but is only capable of a small range of outcomes, being describing behaviour, and modelling data that can be represented through distances between a feature. Other clustering techniques should

be developed to better reflect these features, and should focus on being transparent for better explanations.

5.3. Confidence / Validity

As a project that is focused on areas of research that are still novel, the confidence in the results is by nature low. To maximise confidence, the project focused on adapting known valid approaches onto a new domain. Overall this research can be seen as valid, the results within this project are tangible and effective, but its use must still be verified, meaning the next steps for this project following being submitted will be to explore the effectiveness of this approach on new datasets as appropriate when working with Ai-London.

The use of this approach, on the dataset has proven effective. The clustering has been verified by the decision tree predictive score. The explainable clustering has produced similar groups than is currently being used in industry when tackling a project like this through non machine learning approaches. The labelling of Gone-Aways using this approach cannot be used in certain terms, as it is too general, but can be useful when determining policy as it gives an overall understanding of the behaviours these accounts have. The clusters characterised by Gone-Away traits containing labelled Gone-Aways verify that real patterns of behaviour have been identified.

Overall, this project in the limited scope of this dataset is valid, and it can be extrapolated that there is some validity in other insurance data sets, but this will require further research.

The motor insurance dataset that was tested showed that this approach is not universal. There are limitations on the quality of data that mean that the methods cannot be effective when utilised. This however was limited by the clustering algorithms, which were not new in this research. The explainability approaches still performed, and the fact that they were limited by the clustering methods used shows that with a better algorithm for that dataset, explainability is possible.

6. Evaluation, Reflection, Conclusions

6.1. Project as a Whole

This project went through a lot of changes as it became clear the original objective was not achievable to fit in the restraints of data privacy and time frame proposed. Customer segmentation changed from a preprocessing analysis tool that was an area of interest for understanding Gone-Aways to the main focus of the project, and with that complexities in the structure of the project came. Despite this, a good amount of research has been conducted into how customer segmentation can be used, and an area of research that is currently underdeveloped, explainable clustering, has been expanded upon. Upon reflection, explainable clustering should have been the primary area of research, and more principles from explainable AI should have been carried over.

Data generation was an interesting area of work, but ultimately had little relevance to the work conducted for explainable clustering or for the beneficiary Ai-London, so has been relegated to an appendix for this report. Also, the traceability aspect is an area that will be developed further through working with Ai-London, but is currently underdeveloped. The customer segmentation in this project will be a preprocessing step which has given a better understanding of the customer base. This task requires more data, and different identification services to provide a well rounded research project.

6.2. Research Questions

The purpose of the research questions in this project is to reflect the different areas of study for this project, the customer segmentation, the explainability and the automation. This worked well as it allowed each area to be explored in isolation, and then analysed when combined. It did come with the trade off of being too general; not needling into specific areas of research that this research is desired to be conducted on. This is a reflection of the changing direction of the project over time.

6.3. Literature

The literature for this project is a personal area of interest. With a background in mathematics, it is interesting to read into the algorithms of machine learning models, and see concepts be translated into this field. For the literature review in this project though, perhaps too much focus was put into individual algorithms and adaptations, when only the original versions of the clustering techniques were used.

The literature review did reflect and inform the whole of the project, particularly the research into explainable AI and the principles used. This was a particular area that was identified where the body of research could be improved upon, and justified a large portion of this project.

6.4. Methods

The methodology was written to reflect the decision process behind the approaches and their justifications and was effective in doing this. However, in doing this the section has parts that are too specific to exactly what was done in this report, rather than explaining the design process. That being said, this project contained a wide range of data science techniques, and as such the example implementation was a useful way of explaining how each step is tied together, and is useful as a way to explain the process the script took when automating the process.

6.5. What was Achieved and Implications

In this research, the primary achievement was a proof of concept that customer segmentation with insurance data can be conducted, can be explained and these processes can be done in an automated way. The implication of this being that the techniques used in this project can be constructed into a product that produces customer segments based on data that describes a groups behaviour traits, and that this can be used in industry. The results of this project have reflected a real use case for this research, that being gaining an understanding of Gone-Aways for a CTF policy, and this has provided a real sign that the research will be useful in industry.

Each individual component of the research, the customer segmentation and the explainability approaches, have been evidence that these areas are worth utilising in the insurance industry. Clustering through machine learning is a tool that can be used, and a SOM is a good algorithm for this, but other algorithms should be tested/developed. The explainable approaches are not perfect in this project, but provide real evidence that the principles do apply in this use case. With more developed algorithms, it is possible to have a sophisticated model that can be used for this task.

6.6. Further Work

This project has highlighted many areas that should be researched further. It has already been identified that with the problem of Gone-Aways, that a good tracing algorithm can be effective. Of particular interest personally is in developing a cheap way to trace these accounts, as with a low enough cost this can be used as a public good, and provide benefit to individuals with small amounts in their account where traditional services are not financially reliable.

Further, it could be interesting to develop a better clustering algorithm that is designed for customer segmentation. A large limitation is that customer segmentation requires the use of mixed data types, and more research into these algorithms is required to be able to gain an understanding of categorical data. In this, a distance based algorithm like those used in this project is not wholly suitable, and perhaps a variance based algorithm could perform better.

Explainable machine learning is an interesting area of study. In this area, the focus should transition away from data scientists and towards those that will be using the results of the algorithm. Much of the current body of research relies too heavily on producing explanations within the constraints of machine learning, and it was for this reason one method explored was a statiscal one. More work on explanation after a model is created should be performed, creating explanations that utilise more universal ideas, and concepts that can be easier understood.

References

Albuquerque, P., Alfinto, S., & Torres, C. V. (2012). Support Vector Clustering for Customer Segmentation on Mobile TV Service. *Communications in Statistics - Simulation and Computation*, *44*(6), 1453-1464. 10.1080/03610918.2013.794289

Bayer, J. (2010). Customer segmentation in the telecommunications industry. *Journal of Database Marketing & Customer Strategy Management*, *17*, 247-256. 10.1057/dbm.2010.21

Belle, V., & Papantonis, I. (2021). Principles and Practice of Explainable Machine Learning. *Frontiers in big Data*, *39*. 10.3389/fdata.2021.688969

Cai, F., Le-Khac, N.-A., & Kechadi, T. (2016). Clustering Approaches for Financial Data Analysis: a Survey. *arXiv preprint*. https://arxiv.org/abs/1609.08520

Dasgupta, S., Frost, N., Moshkovitz, M., & Rashtchian, C. (2020). Explainable k-Means and k-Medians Clustering. *Proceedings of the 37th International Conference on Machine Learning*. http://proceedings.mlr.press/v119/moshkovitz20a/moshkovitz20a-supp.pdf

Ding, C., & He, X. (2004). K-means clustering via principal component analysis. *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. 10.1145/1015330.1015408

Hamerly, G., & Elkan, C. (2003). Learning the k in k-means. *Advances in neural information processing systems*, *16*, 281-288. http://papers.nips.cc/paper/2526-teaming-the-k-in-k-means.pdf

Kim, S.-Y., Kim, T.-S., Jung, E.-H., & Hwang, H.-S. (2006). Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert Systems with Applications*, *31*, 101-107. 10.1016/j.eswa.2005.09.004.

Kodinariya, T. M., & Prashant, M. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal*, *1*(6), 90-95. https://www.researchgate.net/profile/Trupti-Kodinariya/publication/313554124_Review_on_Determining_of_Cluster_in_K-means_Clustering/links/5789fda408ae59aa667931d2/Review-on-Determining-of-Cluster-in-K-means-Clustering.pdf

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464-1480. 10.1109/5.58325

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, *36*(2), 451-461. 10.1016/S0031-3203(02)00060-2

Marcus, C. (1998). A practical yet meaningful approach to customer segmentation. *Journal of Consumer Marketing*, *15*(5), 494-504. 10.1108/07363769810235974

ROSCHER, R., BOHN, B., DUARTE, M. F., & GARCKE, J. (2020). Explainable Machine Learning for Scientific Insights and Discoveries. *IEEE Access*, *8*, 42200-42216. 10.1109/ACCESS.2020.2976199

Satopaa, V., Albrecht, J., Irwin, D., & Raghavan, B. (2011). Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. *31st International Conference on Distributed Computing Systems Workshops*, 166-171. 10.1109/ICDCSW.2011.20.

Schmitt, B., & Deboeck, G. (1998). Differential Patterns in Consumer Purchase Preferences using Self-Organizing Maps: A Case Study of China. In *Visual Explorations in Finance* (pp. 141-156). Springer Finance. 10.1007/978-1-4471-3913-3_10

Serrano-Cinca, C. (1998). Let Financial Data Speak for Themselves. In *Visual Explorations in Finance* (pp. 3-24). Springer Finance. 10.1007/978-1-4471-3913-3_1

Sheikh, A., Ghabarpour, T., & Gholamiangonabadi, D. (2019). A Preliminary Study of Fintech Industry: A Two-Stage Clustering Analysis for Customer Segmentation in the B2B Setting. *Journal of Business-to-Business Marketing*, *26*(2), 197-207. 10.1080/1051712X.2019.1603420

Stefanovič, P., & Kurasova, O. (2011). Visual analysis of self-organizing maps. *Nonlinear analysis : modelling and control. Vilnius : Institute of Mathematics and Informatic*, *16*(4). 10.15388/NA.16.4.14091

Tai, W.-S., & Hsu, C.-C. (2012). Growing Self-Organizing Map with cross insert for mixed-type data clustering,. *Applied Soft Computing*, *12*(9), 2856-2866. 10.1016/j.asoc.2012.04.004.

Thinsungnoena, T., Kaoungkub, N., Durongdumronchaib, P., Kerdprasopb, K., & Kerdprasopb, N. (2015). The Clustering Validity with Silhouette and Sum of Squared Errors. *International Conference on Industrial Application Engineering*. 10.12792/iciae2015.012

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11), 2579-2605. https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf?fbclid=IwA

Wang, F., Franco-Penya, H.-H., Kelleher, J. D., Pugh, J., & Ross, R. (2017). An Analysis of the Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity. *International Conference on Machine Learning and Data Mining in Pattern Recognition*, *10358*. 10.1007/978-3-319-62416-7_21

Wu, J., & Lin, Z. (2005). Research on customer segmentation model by clustering. *international conference on Electronic commerce*, *7*, 316-318. 10.1145/1089551.1089610

Yang, W., Long, H., Ma, L., & Sun, H. (2020). Research on Clustering Method Based on Weighted Distance Density and K-Means. *Procedia Computer Science*, *166*, 507-511. 10.1016/j.procs.2020.02.056

Zhao, W.-L., Deng, C.-H., & Ngo, C.-W. (2018). k-means: A revisit. *Neurocomputing*, *291*, 195-206. 10.1016/j.neucom.2018.02.072

Glossary

Customer segmentation - The process of distinguishing groups of customers that fit into similar profiles

Gone-Aways - Used to describe accounts that have not been updated recently and have been inferred to not know about the policy anymore, also known as dormant accounts

Policy holder - The individual that holds the insurance policy

Tracing service - A service that provides the ability to uncover new information about an individual

Dummy/artificial dataset - The dataset that was produced using randomised data modelled on the real data set

Traceability score - The score a policyholder is assigned to represent the likelihood of an individual being identified using a tracing service

Confidence score - A score calculated typically by the tracing services in the confidence a result matches the search

Euclidean geometry - a logical system of rules that defines common-known geometry

Centroids - In this context, the middle point of a K-means cluster. In a general meaning,the centre of plane as calculated from the mean

Hierarchical approach - A top down approach - one that takes the highest level of abstraction first and progressively gets more granular

Customer value - Calculated from the value an insurance gains from keeping the policyholder

Explainable clustering - The methods used that describe the clustering process and give reasons for the clusters produced

Model agnostic explainability methods - An explainable machine learning approach where the model used for the prediction does not matter

Opaque models - machine learning models where the decision process is not evident from the algorithm

Transparent models - machine learning models where the decision process is evident from the algorithm

Sample population - The target data of the test

Parent population - The whole of the data being assessed

Decile - a group of 10% of the data distribution

Appendix A - Calculating the Traceability Score

A key part at the beginning of this project was to quantify how easy it was to trace Gone-Away accounts. For this, an algorithm was used that utilised an external API to search public records and then a weighted similarity score was calculated to generate a score of how confident the individual was to be matched with the result form the API. A machine learning model was built to predict this score. This appendix will briefly explain how this was designed and the results of the methods.

To design this process, it should be understood that the nature of Gone-Away accounts means that the personal information is unreliable. It is uncertain if the address given is still their current address, or if they have moved. This process is designed to identify a likelihood that the individual can be matched on public records. This score is expensive to generate if every account was run on the API, so a machine learning model could be designed to predict this likelihood, if the traceability score is low, it is not worth a search.

The API used was a person search provided by T2A. This API requires the personal information of an individual, name and date of birth, and returns a list of results for results on public records, those being the phone book and open register, of matches. The results include a name, address history and phone number.

The search for this was conducted on 300 accounts, of over 18 accounts, some labelled as Gone-Away and some not. This is because at this point in the project there was not a way developed to identify the likelihood of an account being a Gone-Away. The API search returned an XML file listing all the results, or an error if a result could not be found, or an error if too many results were found. Using this the XML was parsed into a CSV, which allowed for storage of the results as well as analysis. When this was conducted, it was clear there were a substantial amount of errors in the results. These were immediately given a score of 0, indicating no likelihood of a result on the search.

To calculate a score for those that did occur, a weighted mean was calculated. This score was calculated primarily on a similarity check on features like the names, which if strings were compared vectorised the characters first, and the similarity calculation took into account spelling mistakes. The algorithm created also compared the address histories of the policyholders and the API results, the dates being compared, and this was given heavy weighting in the score. Finally, the likelihood score was discounted by a name commonality

value. This value was calculated by comparing how often a name occurs in the KU dataset. The score was designed to be kept between the values of 0 and 1, so that it can be seen as a percentage. For a deeper understanding of this process, the code can be seen.

Once the score was calculated, it could then be interpreted by machine learning models. As this score was divided by scores being 0, errors, and then scores over 0, it made sense to use a classifier first, to identify if a 0 score would be returned. Then, a second model was trained on all results over 0, predicting the likelihood score that was previously calculated. The features this model was trained on are similar to that of the clustering, but included extra features like the commonality of names.

Many models were tested for the regression and classification tasks individually, as can be seen in the jupyter notebook, but the best sores were returned by a Support Vector Machine. This is likely because SVM are proficient at fitting to small quantities of data, like this task has.

The accuracy score of the classification model ws 0.72. The score of the regression model was higher at 0.82. These scores are not as high as might be desired, but do show that the models are predicting the score well, as they are above randomly assigning values to the data points. To improve these scores, it will be necessary to train on more data. Also, it will be necessary to improve the name commonality result. This is because a large proportion of the errors were because of too many results, as a name was too common. When analysing the classification model, name commonality was not a feature that was deemed as important, indicating that the calculation of this score is poor.

Further work in this task would be to first gather more API results to train a better model. It would also be useful to gather more approaches to tracing accounts. Using a similar process as explained here, a score could be calculated for each tracing approach. This would allow for a recommendation system to calculate the cheapest whilst still being an effective way to find an identity for these Gone-Away accounts.

Further, this work could be layered on top of the customer segmentation work of this project. On the surface level, the searches could be limited to just likely Gone-Away accounts identified, giving a precise amount of data. On a deeper level, it could be used to identify the characteristics that underlie a traceable Gone-Away when compared to an untraceable Gone-Away. If this is possible, it could then be identified ways to prevent a Gone-Away from being untraceable.

Appendix B - Data Generation

For this project, a dummy dataset was used heavily. The methods used for this part of the project were statistical, and in this appendix they will be explained, along with explaining the parameters that were tweaked to see how this affected the clustering and explainability approaches in this project. In the data section of this report, it has been touched on the design choices behind the creation of the dataset, but here the process will be fleshed out.

Initially, the design of this dataset was to aid in the production of this report. The purpose was to provide a dataset that was similar to the real data, but one that could be submitted with this report. This was measured in two ways, distribution of the features and covariance between the features. The distribution was replicated so that the make-up of individual features would be replicated. The covariance was theorised to replicate the shared traits between features.

To generate certain features like phone numbers, addresses and names, a library called faker was used. The library requires the input of location. This library allowed for the generation of random addresses that should reflect real places, and random number numbers that replicate the format of a real UK number. Additionally, for features that were of a binary category, namely sex and address, the numpy function random.choice was used. This allows the probability of each option to be set, which allows for the random selection to model the real data.

Replicating the distribution was done by taking metrics from the initial analysis of the real dataset, the mean the standard deviation and the range, and then modelling it as a skewed normal distribution. Using the skewnorm function from scipy, it was possible to parse these metrics and generate a similar spread of random variables to that in the real data set. This was performed on the time series datasets, prior to taking the values like the mean premium payments.

In the initial analysis it was identified that with premium payments there were three broad categories of policy holders. Those that never paid into their accounts, these are the ones that likely were opened by the government and the individual might not have been aware of this creation. There were accounts that paid large amounts into the account a few times, perhaps once a year and then there were accounts that paid into the account an amount on a regular basis, perhaps every month. The probability of each of these was extracted and then modelled using a random choice. A similar process was performed using the address history, identifying that some never moved and some moved frequently.

Once this was done it was identified that the individual features replicated the real dataset and there was a decent replication of the covariance between features as well, as the mean of the sample population for the real and artificial dataset were similar. However, this score was not perfect and it was investigated if this could be improved. To do this, the values were focused on one feature of the data, and then adjusted to better fit the relationships, this was done using the principles of the covariance formula. This however was not an effective technique, it was computationally inefficient, requiring the code to be run on each line rather than each column as before. This resulted in only a smaller dataset being able to be produced in a reasonable amount of time. Further, whilst the covariance between features was more accurate to the real dataset, the distributions became a lot worse, tending to be of a smaller range. In this, it meant that the dataset was better without this step ,and so was not used in this project.

In combination, this process allowed for the generation of a dummy dataset. This dataset did well to replicate the real data, allowing for the testing of code and for an understanding of how well the algorithms tested throughout this project would perform. Predictions using this dataset are not possible, as the data is not real, but it did allow for the development of the methodology for this project.

Throughout this project it has been mentioned that the dataset was manipulated to act as a best case scenario using this data. This was achievable as the methods used in this approach had parameters that could be changed, allowing for the changing of the distributions. For example, to adjust the number of Gone-Aways in the dummy dataset, the random choice probabilities could be increased to be more representative. Alternatively, to change the mean premium payment, the mean value of the skewed distribution could be altered. The benefit of this was that with full control over the distributions, the algorithms could be fully analysed and developed. It was known that the data being used in the clustering was good and distinctions were being produced, so the algorithms should be able to capitalise on this. Further, this allowed for explanations to be planted within the data. The explainable clustering techniques were both distribution based, so it could be set that a large proportion laid outside of the mean. In turn, it allowed for the understanding of whether this was being identified by the explainability algorithms. The methods in this report reflect those that performed best using these methods, increasing confidence in the results of the process.

Appendix C - Code

In this appendix, code extracts will be displayed. For the full code and jupyter notebooks displaying results, see the included extra materials or the github repository. Also contains a full set of results.

Repository

https://github.com/DomPalaczky/Dom-Palaczky-Dissertation-Code

https://drive.google.com/drive/folders/1YsyDVR0CoH8MK27ptsXjuZg5CrCOgwQl?usp=sharing

Tracing

```
def APITracing(data, api_key, sampleSize = None, under18 = False, useDOB = False, useSex=False):

    pernos = []

    if under18 == False:
        #reduce to over 18s
        for idx,row in data.iterrows():
            dob = row['DOB']
            dob = dob.split('-')
            year = str(dob[2])
            if int(year) > 20:
                year = '19' + str(year)
            else:
                year = '20' + str(year)
            if int(year)<2003:
                pernos.append(row['PERNO'])

        datar = data[data['PERNO'].isin(pernos)]

    if sampleSize is not None:
        datar = datar.sample(sampleSize)



    #format DOB
    monthDict = {'JAN': '01',
                 'FEB': '02',
```

```
            'MAR': '03',
            'APR': '04',
            'MAY': '05',
            'JUN': '06',
            'JUL': '07',
            'AUG': '08',
            'SEP': '09',
            'OCT': '10',
            'NOV': '11',
            'DEC': '12'}

resultdict = {}

service_url = 'https://api.t2a.io/rest/'

for idx, row in datar.iterrows():
    forename = row['FIRST_NAME']
    lastname = row['SURNAME']
    postcode = row['POSTCODE']
    if useSex == True:
        sex = row['SEX']
    if useDOB == True:
        dob = row['DOB']
        dob = dob.split('-')
        year = str(dob[2])
        if int(year) > 20:
            year = '19' + str(year)
        else:
            year = '20' + str(year)

        month = monthDict[dob[1]]
        day = dob[0]
    perno = row['PERNO']
    params = {'method': 'person_search',
              'api_key': api_key,
              'forename': forename,
              'lastname': lastname,
              'postcode': postcode}
    if useSex == True:
        params['sex'] = sex
    if useDOB == True:
        params['dob_y'] = year
        params['dob_m'] = month
        params['dob_d'] = day
```

```
        url = service_url +'?' + urllib.parse.urlencode(params)
        http = urllib3.PoolManager()
        response = http.request('GET', url)
        try:
            error =
xmltodict.parse(response.data)['person_search_res']['error_code']
            if error is None:
                details =
xmltodict.parse(response.data)['person_search_res']['person_list']['person']
                resultdict[perno] = details
            else:
                resultdict[perno] = {'error': error}
        except KeyError:
            details =
xmltodict.parse(response.data)['person_search_res']['person_list']['person']
            result = details
            resultdict[perno] = result

    return resultdict

def todf(resultdict):
    l = []

    for resultlist in resultdict.items():
        perno = resultlist[0]
        if type(resultlist[1]) is list:
            for result in resultlist[1]:
                l.append([perno,
                          result['addr_single_line'],
                          result['address_id'],
                          result['address_key'],
                          result['years_text'],
                          result['forename'],
                          result['surname'],
                          result['telephone_number'],
                          result['mobile'],
                          result['person_id']])
        elif type(resultlist[1]) is dict:
            l.append([perno,['error'],resultdict[perno]['error']])
        else:
            result = resultlist[1]
            l.append([perno,
                      result['addr_single_line'],
                      result['address_id'],
                      result['address_key'],
```

```
                    result['years_text'],
                    result['forename'],
                    result['surname'],
                    result['telephone_number'],
                    result['mobile'],
                    result['person_id']])

    df = pd.DataFrame(l)

    df.columns = ['perno','addr_single_line','address_id','address_key',
                  'years_text','forename','surname','telephone_number',
                  'mobile','person_id']

    return df
```

```
def similarity(a,b):
##https://www.datacamp.com/community/tutorials/fuzzy-string-python
    a = str(a)
    b = str(b)
    m = len(a)
    n = len(b)

    dis = np.zeros((m,n))

    for i in range(m):
        for j in range(n):
            if a[i-1] == b[j-1]:
                cost = 0
            else:
                cost = 2

            dis[i,j] = min([dis[i-1,j] + 1,
                            dis[i,j-1] + 1,
                            dis[i-1,j-1] + cost])

    r = ((m+n - dis[i][j]) / (m+n))
    return r

def confidenceScoreCalc(results,
                        pernor,
                        addr,
                        nameWeight = 0.25,
                        surnameWeight = 0.25,
                        numWeight = 0.05,
                        addrWeight = 0.15,
                        addrYrWeight = 0.4,
                        postWeight = 0):
    confidenceScores = []

    for idx,row in results.iterrows():
        #gather data
        if pd.isna(row['address_key']) == True:
            confidenceScores.append(0)
        else:
            p = row['perno']

            add1 = row['addr_single_line']
```

```
            addYr1 = row['years_text']
            name1 = row['forename']
            surname1 = row['surname']
            num1 = row['telephone_number']
            mob1 = row['mobile']

            if pd.isna(add1) == False:
                post1 = add1.split(', ')[-1]
            else:
                post1 = 'NaN'

            pernod = pernor[pernor['PERNO'] == p]
            addd = addr[addr['PERNO'] == p]

            add2 = list(addd['ADDRESS'])
            post2 = list(addd['POSTCODE'])
            datesa = list(addd['DATE_FROM'])
            datesb = list(addd['DATE_TO'])
            addYr2 = []
            for i in range(len(datesa)):
                d1 = str(datesa[i]).split('-')[-1]
                if int(d1) > 25:
                    d1 = '19' + d1
                else:
                    d1 = '20' + d1

                d2 = datesb[i]

                if pd.isna(d2) == True:
                    addYr2.append(d1 + '-')
                else:
                    addYr2.append(d1 + '-' + str(d2))

            name2 = list(pernod['FIRST_NAME'])[0]
            surname2 = list(pernod['SURNAME'])[0]
            num2 = list(pernod['HOME_PHONE_NUMBER'])[0]
            mob2 = list(pernod['MOBILE_PHONE_NUMBER'])[0]
            tele2 = list(pernod['WORK_TELEPHONE_NUMBER'])[0]

            #run similarity chacks
            nameScore = similarity(name1,name2) * nameWeight
            surnameScore = similarity(surname1,surname2) * surnameWeight

            numSims = []
            i=0
```

```
            if pd.isna(num1) == False:
                if pd.isna(num2) == False:
                    numSims.append(similarity(num1,num2))
                    i+=1
                if pd.isna(mob2) == False:
                    numSims.append(similarity(num1,mob2))
                    i+=1
                if pd.isna(mob2) == False:
                    numSims.append(similarity(num1,tele2))
                    i+=1
            if pd.isna(mob1) == False:
                if pd.isna(num2) == False:
                    numSims.append(similarity(mob1,num2))
                    i+=1
                if pd.isna(mob2) == False:
                    numSims.append(similarity(mob1,mob2))
                    i+=1
                if pd.isna(mob2) == False:
                    numSims.append(similarity(mob1,tele2))
                    i+=1

            if i > 0:
                numScore = np.max(numSims) * numWeight
            else:
                numScore = 0

            addsims = []
            for a in add2:
                addsims.append(similarity(add1,a))
            addScore = np.max(addsims) * addrWeight

            postsims = []
            for a in post2:
                postsims.append(similarity(post1,a))
            postScore = np.max(postsims) * postWeight

            yrsims = []
            for a in addYr2:
                yrsims.append(similarity(addYr1,a))
            yrScore = np.max(yrsims) * addrYrWeight

            numResults = len(list(results[results['perno'] ==
p]['perno']))**0.1

            sim = (nameScore + surnameScore + numScore + addScore + yrScore
```

```
+ postScore) / numResults

            confidenceScores.append(sim)

    return confidenceScores
```

Policy Holder history

```
def generateHistory(no):
    m = np.random.choice(3, p = [0.8, .15,.05]) + 1

    prevaddre = []
    prems = []
    prevaddre = [[no, fake.address().replace('\n',', '),
(fake.date_between('-18y','today'))] for i in range(m)]
    chnce = random.uniform(0, 1)
    if chnce < 0.33:
        prems = [[no, abs(round(skewnorm.rvs(10, loc=20, scale=100),2)),
str(i)[0:10]] for i in pd.date_range('2003-01-01','2021-01-01',freq =
'm')]
    elif chnce > 0.66: # if irregular, generate random dates the prems
are paid, and random prem amounts
        n=random.randrange(1,100)
        prems = [[no, abs(round(skewnorm.rvs(0, loc=2000,
scale=1000),2)), (fake.date_between('-18y','today'))] for i in range(n)]
    else:
        d = (datetime.now() - relativedelta(years=18) +
relativedelta(days = np.random.randint(-100,100))).date() #18 year olds +
some jitter
        prem = 250
        l2 = [no,prem,str(d)]
        prems.append(l2)

    return prevaddre, prems
```

Policy Value

```
for no in pernos:
    if no in premsdf.index:
        nowval = []
        #craete lists to iterate through
        p = premsdf.loc[no]['netPrem']
        d = premsdf.loc[no]['effDate']

        if str(type(p)) == "<class 'pandas.core.series.Series'>":
            lp = list(p)
            ld = list(d)
        else:
            lp = [p]
            ld = [d]

        for i in range(0,len(lp)-1):
            #calc differnce in dates by years
            effd = datetime.strptime(str(ld[i]) ,'%Y-%m-%d')
            yearDiff = relativedelta(edate, effd).years
            #calculate value now with interest
            interest = lp[i]*(1.04**int(yearDiff))
            nowval.append([interest,yearDiff])

        #Assume only one input payment of 250 (plus interest = 350), calc
fees

        totalval = 710
        prevyear = 18

        for i in nowval:
            if prevyear == i[1]-1:
                totalval = totalval * 0.99 #take fee
            totalval = totalval + i[0] #add prem
            prevyear = i[1]

        totalvals.append(totalval)
    else:
        totalvals.append(595)
```

```
    def findKMeansClusters(self, x, path, minClusters=2, maxClusters=50):
        #this runs through up to maxClusters to identify the optimal
number of clusters according to the kneedle technique

        clusters = []
        #scores = []
        inertias = []

        #nomrmalise x
        x = MinMaxScaler().fit_transform(x)

        #run kmeans on cluster sizes
        for i in range(minClusters, maxClusters):
            kmeans = KMeans(n_clusters=i, random_state=0).fit(x)
            clusters.append(i)
            #scores.append(kmeans.score(x))
            inertias.append(kmeans.inertia_)

        #normalise
        intertiasNorm = self.Norm(inertias)

        #plot graph
        ax = self.plotInertia(clusters, intertiasNorm, path)

        kneedle =
kneed.KneeLocator(np.arange(np.array(intertiasNorm).size), intertiasNorm,
curve = "convex", direction="decreasing")

        self.knee = kneedle.knee

        return inertias, kneedle

    def labelKMeans(self,x, maxclusters = 10, clustersNum = 5, useKneedle
= True):
        #this labels a dataset given k clusters, requires a knee

        if maxclusters < self.knee: #if knee is over mac clusters, use
max clusters
            useKneedle = False
            clustersNum = maxclusters
```

```
            print('Warning: too many kneedle clusters, reverting to max
clusters')

        if useKneedle == True:
            clustersNum = self.knee

        x = MinMaxScaler().fit_transform(x) #scale data

        kmeans = KMeans(n_clusters=clustersNum, random_state=0).fit(x)
#run Kmeans

        #find labels
        labels = kmeans.predict(x)
        centers = kmeans.cluster_centers_

        return labels, centers, kmeans
```

```
    def somTrain(self, X, param):
        #given param dict train a SOM
        som = MiniSom(x = param[0], y = param[0], sigma = param[1],
learning_rate = param[2], neighborhood_function = param[4], input_len =
X.shape[1], random_seed = 1)
        som.train(X, num_iteration = param[3])

        return som.quantization_error(X), som.topographic_error(X)

    def train_som(self, X, params = None, returnBestTrained=True):
        #perform a grid search of given parameters, then returns lowest
error
        if params is None:
            params = {'xy': [5,8,10,12,15,20,25],
                      'sigma' : [0.5,0.75,1,2,3,4],
                      'learning_rate' : [0.2,0.5,0.75,1,2,5],
                      'iterations' : [10000],
                      'neighborhood_function' : ['gaussian']}

        paramsCombined = list(it.product(*(params[p] for p in params))) #
create a combination of all parameters

        results = {p:self.somTrain(X, p) for p in paramsCombined} #
dictionary of results of all params

        best = list(sorted(results.items(), key=lambda item:
item[1]))[0][0] #lowest quat error
        if returnBestTrained == True:
            som = MiniSom(x = best[0], y = best[0], sigma = best[1],
learning_rate = best[2], neighborhood_function = best[4], input_len =
X.shape[1], random_seed = 1)
            som.train(X, num_iteration = best[3])
            print("The best parameters are: " + str(best))
            return results, best, som
        else:
            return results, best
```

```
    def SOMKmeans(self, som, path, returnLabelsShaped = False):
        # use K-means to aggregate clusters
```

```
        weights = som.get_weights()
        weightsshape = np.shape(weights)
        weightsreshape = weights.reshape(-1, weights.shape[-1])

        km = kmeansCluster()
        inertias, kneedle = km.findKMeansClusters(weightsreshape, path =
path)

        labels, centers, kmeans = km.labelKMeans(weightsreshape)

        if returnLabelsShaped == True:
            labels = labels.reshape(weightsshape[0],weightsshape[1])

        return labels

    def SOMKmeansWinners(self, som, X, labels):
        #creates Kmeans clusters based on SOM labelling
        winners = [list(som.winner(x)) for x in X]

        kmeanslabels = [labels[w[0],w[1]] for w in winners]

        #some neurons don't win, meaning some k means clusters are empty
        labeldict={}
        i=0
        for l in np.unique(kmeanslabels):
            labeldict[int(l)] = i
            i += 1

        kmeanslabels = [labeldict[l] for l in kmeanslabels]

        return kmeanslabels
```

```
def featureImportanceStats(clustersummarydf, numofclusters, X):

    importance = {key: [] for key in range(numofclusters)}
    impvals = []
    i=0
    for c in clustersummarydf.columns:

        l = clustersummarydf[c]
        x = X[:,i]

        for j in range(len(l)):
            if l[j] > np.mean(x) + (np.std(x)): # test representativeness
                impvals.append([c,j,np.mean(x), 'high']) #column,
cluster, mean value, high or low
            elif l[j] < np.mean(x) - (np.std(x)): # test
representativeness
                impvals.append([c,j,np.mean(x), 'low'])

        out = ([ i for i in range(len(l)) if l[i] > np.mean(x) +
(np.std(x)) or l[i] < np.mean(x) - (np.std(x))])

        for clus in out:
            importance[clus].append(c)

        i+=1
    # for key in list(importance.keys()):
    #     print('Cluster ' + str(key) + ' has important features: ' +
str(importance[key]))

    impvals.sort(key=lambda x: x[1])
    print('Statistically')
    for v in impvals: #print important clusters
        print('Cluster ' + str(v[1]) + ' has relatively ' + str(v[3]) + '
feature ' + str(v[0]) + ' with mean value of ' + str(v[2]))


    return importance, impvals

def featureImportanceDT(X,labels, returnScore=False):
    clf = DecisionTreeClassifier(random_state=0, max_depth = 5)
```

```
    clf.fit(X, labels)
    if returnScore == True:
        score = cross_val_score(clf, X, labels, cv=5)
        return clf.feature_importances_, score, clf
    else:
        return clf.feature_importances_, clf

def plotFeatureImportance(columnnames, featureImportance, path, overall =
True, clusnum = None, show=False):
    #create directory
    try:
        os.mkdir(path + '/feature_importance')
    except:
        pass

    if overall == True:
        fig1, ax1 = plt.subplots()
        ax1.bar(columnnames[:], featureImportance)
        plt.setp(ax1.get_xticklabels(), rotation=30,
horizontalalignment='right')
        plt.title('Overall Feature Importance')
        fig1.savefig(path +'/feature_importance/feature importance.png',
transparent=False, facecolor='white')
        plt.clf()
    else:
        plt.bar(x = columnnames, height = featureImportance)
        plt.title('Feature importance for cluster ' + str(clusnum))
        lo, la = plt.xticks()
        plt.setp(la, rotation=30, horizontalalignment='right')
        plt.savefig(path +'/feature_importance/feature importance cluster
' +str(clusnum) +'.png', transparent=False, facecolor='white')


        if show == True:
            plt.show()

        plt.clf()
```

Appendix D - Project Proposal

Project Proposal – How can Machine Learning Techniques be Applied to Customer Segmentation and Identifying CTF Policy Holders?

1. Introduction

This project will focus on Child Trust Fund (CTF) gone-aways. A child trust fund was a long-term tax-free savings account for children, that reaches maturity when the child turns 18. The first accounts were created for children born in 2003, and as such they have begun to reach maturity. This has highlighted the issue of gone-aways, as many have not claimed their accounts. A gone-away is identified by an individual who has either left or forgotten their CTF account, but it remains open. It is estimated that the value of the accounts of these gone-aways is as much as £3 billion, showing the full extent of this problem. In this project, a combination of machine learning techniques will be used to characterise and then find potential gone-aways, allowing for a better understanding of who and where these people are.

The data for this project will primarily come from Kingston Unity. This was a life company that prides itself on its social responsibility, as a mutual their members in part own the business. In the case of gone-aways this is important, as there is little incentive for the average company to encourage individuals to withdraw their money from accounts; a company's profits are directly related to the value of their liabilities. The data contains sensitive financial info so the data submitted will be obfuscated/censored. This can have impact on the meaning of the results, so this research project will focus on a transferable methodology.

In this data, there are labelled gone-aways. These are manually inputted and are individuals who have not replied to some sort of communication. It can also be assumed that an individual that has not updated their account or input any premium is a gone away. The data used will be on these cases. The issue with gone-aways is that the currently held data is incorrect, their contact details could be out of date, meaning attempts to contact them will not work, so an identification service is required. It is expensive to use these services, so the best service must be identified.

This project is being conducted as part of an internship for Ai-London and as such is guided by what the client and in turn Ai-London desire and recommend. There are certain approaches and

models that have been recommended to tackle this problem, but the research, methodology, and results will be of my own work.

The approach to this problem will be split into two distinct segments. The first will be building a clustering model for customer segmentation. This is an unsupervised approach that will group CTF holders into distinct categories. This model will create a structure for the data, as well as a generalised understanding of the individuals. Once this has been performed a recommendation system will be applied to the clusters, to identify the best way to find information on the gone-aways. This will likely be a recommendation of a third-party tracing system. These tracing services could be Experian, Id3 Global.

The first objective of this project is to develop a meaningful way to segment and characterise gone-aways. The main approach for this will be unsupervised clustering. The second objective is to use these clusters to recommend the best tracing system. This is a regression task and explores how the segmentations are useful in application. The research questions are: How can unsupervised machine learning customer segmentation be applied to insurance data and CTF gone-aways? How can a regression model be used to predict identification service confidence scores? Can a combination model of clustering and regression be used to identify CTF gone-aways?

The outcome of this project will be a method of characterising policy holders and then identifying ways of tracing individuals. This methodology will be useful, as it can provide a part of the solution to the overall problem of gone-aways, as well as a deeper understanding of how to segment these customers. For research interest it will provide a combination model for using clustering in recommendation systems and an application for this approach. The results will be explainable segments of customers, applied to a recommendation system for finding information on these gone-aways, this can then be used by AI-London or Kingston Unity to find the best way of contacting these individuals.

2. Critical Context

Customer segmentation has been a popular research area. This is mostly in retail, where the amount of data is vast and the benefits easy to see, typically to maximise profit. From there, other industries have begun using customer segmentation, each providing their own challenges in identifying what data is of most importance, and how to use the segments once performed. In this

review [1] there is an overview of how customer segmentation is currently performed, as well as the different machine learning methods being used. This review forms a good baseline for further study into this area. The review provides 4 approaches. In the project, an altered version of the activity based technique will be applied, using an account holders history to group.

As an explanation for K-means in this application, this paper [2] provides a comparison to other traditional customer segmentation techniques, finding that this simple algorithm performs well.. The elbow method is applied to find the optimal number of clusters. The approach here was to take two features, apply scaling and then examine how the algorithms performed. The analysis was visual, examining the distance between clusters.

Clustering techniques have been applied to the finance industry. This paper [3] explains a technique for clustering analysis customer segmentation in business to business. The paper utilises K-means, providing an ad-hoc approach to labelling segments. Features like the regularity of customers, to produce clusters which it then uses metrics to assess the quality of customers, grouping them into categories like ‘young and churned’ or ‘loyal and valuable’. Some of these metrics can be applied in this research project; it is hoped that the segments produced will be as explainable.

Other machine learning approaches have been applied to customer segmentation in the finance industry. Smeureanu et al [4] provide a way to use more advanced machine learning approaches, support vector machines and artificial neural networks to create meaningful segments of customers. Using classification, the researchers separated groups depending on how affluent the customer was. As in this project, the data is unlabelled, a similar approach cannot be used at the clustering stage. The research paper does provide an interesting insight into how the data could be prepared for the regression model that will be produced. In particular, the average of transactions and the counts. A neural net could be used in the form of a self-organising map (SOM) as in this paper [5]. Here, two levels of abstraction are applied the first creates a large number of groups and the second clusters these groups, providing the final segments. This technique can be applied to the data in this project.

Of particular interest in this research is how these classifiers were applied to create segments of customers. Exploring this deeper, Albuquerque et al [6] provide a more advanced method of obtaining clusters from support vector machine. Applying the Support Vector Clustering (SVC) proposed by Ben-Hur et al, to create internally homogenous groups. Using this, the researchers created explainable groups, highlighting the features that make each group unique. From here they highlight how policy changes could affect different groups. It is also highlighted that the SVC is well performing as it performs well with arbitrary geometry, indicting it will work well in the project.

3. Approach

The approach for this project is primarily split into two parts. The focus of this project is to use machine learning tools, primarily clustering, to segment policy holders with the goal of achieving a deeper understanding of them. This will in turn be useful for a multitude of things, but of particular interest in this project is to find those gone-aways. So, the approach to this problem can be seen as focusing on customer segmentation, with an application in finding gone-aways.

For the purpose of this project that will be submitted, there are data security concerns which limit the amount of data that can be provided, as such, much of the data will be censored/obfuscated. This project is being conducted as a part of an internship with Ai-London, and as such will need to be transferable to real data too. This will need to be considered throughout the project, and ways to mitigate will be mentioned.

The first stage of this project is feature selection. The goal of this stage is to find the features that will provide a good representation of the data. It is also important to use a minimal amount of data, the base dataset is very large, it will be easy to include too much information, which will lead to inefficient model building. The primary areas of consideration here are: geospatial, temporal transactions, transactions, value of policy, personal information. All these areas will require individual analysis and transformation prior to being used to train the model. For all these data points, some sort of obfuscation will be needed to be made for the submission, to ensure that data privacy rules are kept to. This can be done in many ways, like scrambling an individual's name and numbers to representing an address in a different way, and will have minimal effect on results.

After the features have been selected and pre-processed, an initial analysis will be performed. In this stage the aim is to create a basic analysis of the data. This will help get an initial understanding of the data, what is possible and if any further pre-processing may be required. It is also at this stage there will be a clearer idea of which modelling techniques will best be used for the clustering. The initial analysis will allow for a more detailed plan of to implement the machine learning techniques. At this point, the data will be separated into two categories, known gone-aways, those who have been manually flagged and not known gone-aways.

Now the clustering techniques can be applied. The goal here is to use multiple different models and compare how they perform on the dataset. The clusters will represent something meaningful in the data, for example, a 'dormant and unaware' segment for those who are not aware they hold a CTF, and are not actively adding money. Some models that will be used are more basic algorithms like

K-means, or KNN as well as more complex sampling techniques, like a deep neural network or Support Vector Machine. The outcome of this stage will be multiple different models that should produce a collection of samples that represent a group of customers with similar attributes. The models will be assessed and reduced to a couple that have the most confidence of providing meaningful clusters. The models will be able to handle multivariate data. At this stage, the performance of the models will be quantified, as well as calculating the time and resources used in creating the clusters. This is important, as this project could be used commercially on large datasets, and as such requires particular interest in the cost of running the algorithms.

Once appropriate segments have been created, they will be tested to find their meaningfulness and utility, by finding the gone-aways. To do this, a recommendation system will be created to identify the identification service that is likely to produce the best confidence score for an individual. This will be a regression model, used to predict the confidence score a particular service provides. The scores can then be ranked, and a recommendation can be made. The utility of this for this project will be that if the clusters aide in producing a well performing classifier, they are valid, and therefore can be said to represent a meaningful segment of customers. For Ai-London, this will provide a product that can be used to recommend how to identify gone-aways. In this stage, the obfuscated data provides more of an issue, so the confidence score is used instead of identifying personal information.

The methodology is what is most important in this project. Whichever models are used must be transferable to different data sets, as this project will be based around both obfuscated and real data, but also so that there is a product produced at the end for Ai-London that is transferable onto other datasets.

The primary limitation of this approach is the use of obfuscated data. This is necessary for data security but requires mitigation throughout the project so that the data can both be kept private and be suitable for this project. The methodology for this project must be transferable to real and obfuscated data.

Another limitation of this project is having no clear ensuring the customer segments are meaningful. Using measures like silhouette score can inform of how separate the clusters are, with the idea of more separate clusters are likely to carry more meaning. Further, with the application of the segments onto the task of finding the best identification service will allow for a level of confidence that the segments are valid.

The final thing to be cautious of is keeping the project separate from the other interns at Ai-London. In particular, one who is using the same dataset as I. The project being undertaken here is purposefully different to his. The focus on methodology, and the clear separation in research areas will allow for all parts of this project to have a clear distinction. Further, Ai-London have been informed of this necessity, and will be aiding in keeping the work relating to this research separate.

4. Work Plan

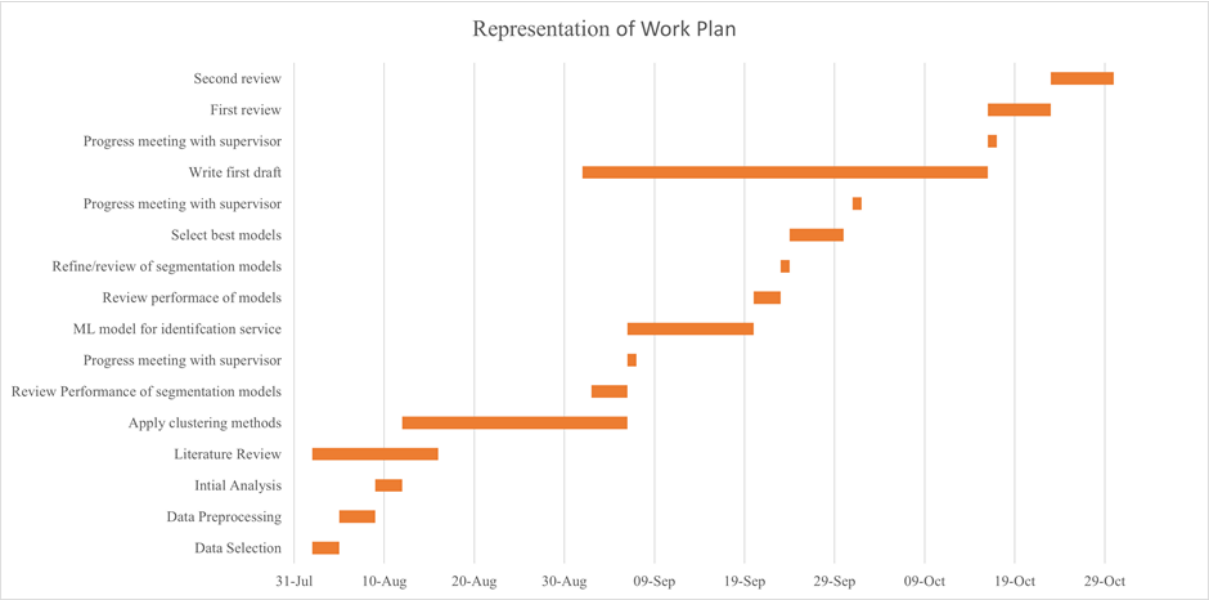

Figure 1: Gantt Chart for Work Plan

5. Risks

Risk	Impact	Likelihood	Mitigation

Data security restricts data	High	Low	Prior checks and measures taken
Not enough time at internship to complete project	High	Low	Well thought-out workplan
Models take a long time to train	Med	Low	Planned to have models trained early
Sickness (i.e. Covid 19 / vaccine side effects)	Low	Low	Delay work plan by a few days
Ai-London / client needs change	High	Low	Apply methods in this project to another situation
Coding is lost/corrupted	High	Low	Back up on repository site
Overlap of project with another intern	High	Low	Measures taken throughout this plan to ensure this does not happen

References

[1] Bruce Cooil, Lerzan Aksoy, Timothy L. Keiningham, Approaches to Customer Segmentation, 2008, Journal of Relationship Marketing, doi:10.1300/J366v06n03_02.

[2] Tushar Kansal, Suraj Bahuguna, Vishal Singh, Tanupriya Choudhury, Customer Segmentation using K-means Clustering, 2018, International Conference on Computational Techniques, Electronics and Mechanical Systems, doi: 10.1109/CTEMS.2018.8769171.

[3] Alireza Sheikh, Tohid Ghanbarpour, Davoud Gholamiangonabadi, A Preliminary Study of Fintech Industry: A Two-Stage Clustering Analysis for Customer Segmentation in the B2B Setting, 2019, Journal of Business-to-Business Marketing, doi:10.1080/1051712X.2019.1603420.

[4] Ion Smeureanu, Gheorghe Ruxanda, Laura Maria Badea, Customer segmentation in private banking sector using machine learning techniques, 2013, Journal of Business Economics and Management, doi:10.3846/16111699.2012.749807.

[5] J. Vesanto, E. Alhoniemi, Clustering of the self-organizing map, 2000, IEEE Transactions on Neural Networks, doi: 10.1109/72.846731.

[6] Pedro Albuquerque, Solange Alfinito, Claudio V. Torres, Support Vector Clustering for Customer Segmentation on Mobile TV Service, 2015, Communications in Statistics - Simulation and Computation, doi:10.1080/03610918.2013.794289.

Research Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

http://www.city.ac.uk/department-computer-science/research-ethics

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part.
The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk.

The approval may be ***provisional*** – *identifying the planned research as* likely to involve MINIMAL RISK.

In such cases you must additionally seek ***full approval*** from the supervisor as the project progresses and details are established. ***Full approval*** must be acquired in writing, before beginning the planned research.

Part A: Ethics Checklist

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)? e.g. because you are recruiting current NHS patients or staff? If you are unsure try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation?	**NO**

	Such research needs to be authorised by the ethics approval system of the National Offender Management Service.	
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about special category or sensitive subjects? *For example, but not limited to: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	**NO**

2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? These may include, but are not limited to, electrical stimulation, heat, cold or bruising.	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ **Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.**		*Delete as appropriate*
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.	**NO**

3.3	Are participants recruited because they are staff or students of City, University of London? For example, students studying on a particular course or module. If yes, then approval is also required from the Head of Department or Programme Director.	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data?	**NO**

	For example, as interviewees, respondents to a survey or participants in testing.	