Development and Research of Methods to Classify Houses in Multiple Occupation

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science in **Data Science**

Under the supervision of

at the

University of Salford School of Computing, Science and Engineering

September 2019

University of Salford MANCHESTER

Abstract

The number of houses in multiple occupation (HMOs) identified in Bedfordshire has increased and this type of accommodation is frequently occupied by vulnerable residents, those on a minimum wage or less economically active than the national average person who are owner occupiers. Central Bedfordshire Council (CBC) wish to develop new methods for identifying these types of properties. In this project, a new and scalable method of identifying HMO properties using rent deposit records was developed. The method uses approximate string-matching techniques and hierarchical centroid-linkage clustering (HCLC) to detect properties with multiple occupants. HCLC was conducted on unique pairs of addresses derived from rent deposit records to form clusters of addresses which refer to the same property, in order to find properties with more than 3 occupants and performed with an average precision of 91.95 $\pm$ 2.15%, detecting 64 potential HMO properties. The total number of HMO properties detected was also consistent with the expected number of HMO properties, based on analysis of the 2011 UK census and English Housing Survey of 2008. Further research was conducted to assess the feasibility of developing a rare-event classification model to detect fraudulent property advertisements, where so-called 'rogue-landlords' covertly advertise HMO properties while avoiding appropriate licensing. A prototype recursive partitioning classification model developed showed that, while performing with high accuracy, property advertisements typically do not contain enough predictors which highly correlate with HMO advertisements to reliably discriminate between fraudulent and non-fraudulent advertisements and it was concluded that the high accuracy was indicative of overfitting.

Contents

List of Figures

List of Tables

1 Introduction

There are currently 11,000 households privately renting in Bedfordshire with further planned growth of 39,350 dwellings between 2015-2035. Large companies such as Amazon and Superdrug have moved into the area in the form of industrial warehouses which have created 1000s of new jobs for residents. However, the corresponding impact of this has been individuals and families renting properties based on cost and not on conditions. The number of Houses in Multiple Occupation (HMOs) identified in Bedfordshire has increased and this type of accommodation is frequently occupied by vulnerable residents, those on a minimum wage or less economically active than the national average person who are owner occupiers. Central Bedfordshire Council (CBC) has identified an increasingly steep rise in the number of HMOs in the last 4 years, from 51 to 119. Basic data processing by CBC using rent deposit data has identified an additional 180 HMOs not known to the Council. CBC wants to pro-actively identify further HMOs and poor-quality privately rented accommodation in the district.

The concept of HMOs was first described in section 345 of the Housing Act 1985, which defined a HMO as "a house which is occupied by persons who do not form a single household" [6]. The definition of a HMO was later refined to a dwelling or section of a dwelling (such as an enclosed living quarters like a flat), which is occupied by more than one household. The legal definition is now outlined in sections 254-260 of the 2004 Housing Act [7]. This act also introduced the requirement for some HMOs to be licensed. The key attributes of a HMO include: houses and flats that are shared by more than two people who do not have a familial relationship, Properties which have been sub-setted into enclosed, individual dwellings but with some sharing of facilities, such as a bathroom or communal areas; part-converted properties that contain one or more sub-setted private living spaces, property conversions which have been transformed into self-contained living spaces but which were not converted to the standard set by the Building Regulations 1991 and where the owner of the property occupies less than $\frac{2}{3}$ of the property area. A HMO that requires a mandatory license is prescribed under section 55 of the 2004 Housing Act and The Licensing of Housing in Multiple Occupation (Prescribed Description) (England) Order 2018. The definition is:

(a) a dwelling which is occupied by five or more persons;

(b) is occupied by persons living in two or more separate households; and

(c) meets:

(i) the standard test under section 254(2) of the Act;

(ii) the self-contained flat test under section 254(3) of the Act but is not a purpose-built flat situated in a block comprising three or more self-contained flats; or

(iii) the converted building test under section 254(4) of the Act.

1.1 Motivations

In some circumstances HMOs provide a pragmatic solution to the privately rented housing sector's needs in offering accommodation to university students and young professionals who wish to live in densely populated town and city areas they would not otherwise be able to afford. Conversely, individuals who are unable to attain traditional forms on tenure may resort to HMOs, resulting in an increased incidence of vulnerable residents suffering from poor living conditions inflicted by 'rogue landlords' [8]. Vulnerable residents commonly noted in literature include: homeless people, ex-offenders recently released from prison, young adults leaving the social care system, such as orphans; people with mental health illnesses and controlled drug abusers [9]. Laylard noted that HMOs occur in concentrated areas and give rise to a intense change in the character of the neighbourhood in which they occur. This in turn brings about an increased incidence of complaints by local residents regarding noise pollution, problems with rubbish collection, parking difficulties and building maintenance [10]. Furthermore, HMO properties are frequently poorly managed and bring about a number of safety concerns including greater risks of energy poverty, fires, damp, mould and injuries due to falls between levels [11].

The Housing Solutions Service at CBC is responsible for enforcing a wide range of statutory provisions relating to housing and environmental conditions affecting health and safety. This includes: improving the standards, condition and quality of housing in Bedfordshire;

reducing the number of properties with serious risks to health and safety, reducing the number of vulnerable households living in non-decent homes, improving the energy efficiency of properties, reducing the rate of fuel poverty in homes and improving the standards in HMOs [12]. A 2016 study by Cauvain et al found that there was a substantial under-representation of HMO properties in English housing datasets. They reasoned that housing surveys tend to disregard or under-represent HMO properties through poor survey design and sampling techniques. They also remarked that because of poor survey design and sampling techniques, HMO properties are under-represented in large population surveys such as the UK census and that landlords and tenants of HMOs benefit from this because they can avoid detection from authorities [13]. The Housing and Planning Act 2016 introduced the option for local authorities to impose a financial penalty on an individual or organisation as an alternative to prosecution for certain housing offences under the Housing Act 2004 [14]. In the CBC area, failure to obtain a HMO license or breaching HMO management regulations can result in a financial penalty of up to £30,000, thus making the detection of HMOs a lucrative undertaking for CBC and other local authorities [15].

1.2 Problem Description and Objectives

HMOs present themselves as a rare-event classification problem. There are differing estimates for the number of HMO properties reported in literature. For example, the English Housing survey of counties performed in 2008 estimates that there are around 236,000 HMOs in England, approximately 1.1% of privately rented properties; whereas analysis of the 2011 census data report by Cauvain estimates that HMO properties make up approximately 4.3% of privately rented properties [16, 9]. If the CBC area is consistent with these estimates, then we can expect HMO properties in Bedfordshire to be within the range of 1.1-4.3% of the total number of private rented sector properties and this will manifest in the form of imbalanced datasets. Developing learner models from data containing instances or classes which are substantially rare using standard learning algorithms significantly compromises the predictive accuracy for the rare class [17]. Therefore methods for rare-event classification problems should be adopted in order to achieve good generalization performance.

The council holds data on the number of rent deposits for a given property, counting these would indicate how many tenants are living at the property. Analysis of rent deposit data is a valid approach to determine whether a property is a HMO because the legal definition of a HMO is predicated on a property's number of tenants, however; it raises some systematic error because rent deposits do not specify whether the tenants have a familial relationship. The aim of the project is to provide a decision support framework which can be reviewed by professional Housing Officers at CBC, so systematic errors such as these do not need to be fully accounted for, instead the framework should provide corroborating evidence for properties which are suspected HMOs or used to direct Housing Officers to potential HMOs. Rent deposit data held by the council comes in the form of semi-structured text data with an inconsistent schema, which designates property addresses. String distance functions or methods of text processing will be required to cluster rent deposits to their associated addresses.

Two core research objectives (RO) were formulated for this project, along with 2 sub-objectives. The first research objective is to develop the decision support framework for Housing Officers at CBC and is as follows:

Research Objective 1: *Develop a method of clustering addresses which refer to the same property from rent deposit data to reflect a properties HMO status.*

The majority of private rented sector landlords provide well managed accommodation, but there are a small number of so-called 'rogue landlords' and property agents who knowingly breach the law and rent out HMO accommodation illegally. This is sometimes achieved by covertly advertising HMO properties on real estate portal websites such as Rightmove, these properties are referred to as "non-compliant HMOs". The second core research objective will focus on developing a prototype method of detecting HMO properties which doesn't rely on data local authorities collect.

Research Objective 2: *Conduct research to test whether it is feasible to develop a classification model for predicting the likelihood of a property advertisement being a HMO.*

RO2.1 *Introduce candidate features for the classification model which are easily accessible from property advertisements and correlate highly with the likelihood of being a HMO.*

RO2.2 *Create an original dataset containing instances of a HMO and non-HMO target variable along with predictors for the creation of training and validation sets.*

As part of CBC's Housing Enforcement Policy, CBC proposes to operate a programme of inspections for any HMOs discovered in Central Bedfordshire. Misclassifying a property as a HMO could potentially waste council time and resources, so both core research objectives will be principally evaluated by their classification precision. In addition to this, RO1 must also be evaluated with respect to its usability as a decision support framework for Housing Officers.

1.3 Research Approach

The Cross-Industry Standard Process for Data Mining (CRISP-DM) was adopted for this project. CRISP-DM is a data mining methodology, described in terms of a hierarchical process model (Figure 1). CRISP-DM was conceived first in 1996; it was designed to be an industry-neutral, standardized data mining framework to address issues with project replication and has since become the most frequently adopted and most cited data mining methodology in literature [1, 18]. CRISP-DM describes the lifecycle of a data mining project in 6 phases. The 6 phases are listed here and have been adapted from Wirth et al [19].

1. **Business Understanding**: In this phase, the project objectives are established, which are converted into a data mining problem. This phase is closely linked to the data understanding phase, business understanding is often required to make sense of the available data.

2. **Data Understanding**: this phase involves initial data collection, it entails exploratory data analysis and assessments of data quality in order to develop precursory insights into the data.

3. **Data Preparation**: this phase includes all activities to construct the final dataset

which is used in the modelling phase. Tasks include, attribute selection, data cleaning and construction of new attributes.

4. **Modelling**: in this phase modelling techniques are selected and applied, further preliminary testing and revision are also involved.

5. **Evaluation**: evaluation of model with respect to business objectives. A key objective of this phase is to determine if there are issues with modelling that have not been sufficiently considered and whether said issues affect the performance of the model.

6. **Deployment**: depending on the requirements, the deployment phase involves generating a report or implementing a repeatable data mining process.

This project requires significant business understanding to conduct meaningful analysis of data, without knowledge of the relevant legislation and council policies, it would be difficult to formulate appropriate research objectives. CRISP-DM makes considerations for business understanding, whereas other popular data mining methodologies such as "Knowledge Discovery in Databases" (KDD) and "Sample, Explore, Modify, Model, Assess" (SEMMA) do not [20]. In this project, the 'Business Understanding' and 'Data Understanding' phases of the CRISP-DM methodology have been expanded to make considerations for the theoretical underpinnings of the project. The problem description explained that HMO properties present themselves as a rare-event classification problem, so special considerations will be required to develop models classifying them. Therefore, 'Business Understanding' in this context will also include a review of literature to understand how to model rare-events appropriately. The 'Evaluation' phase will correspond to the use of performance metrics establish in section 1.2 and whether the work done fulfils the requirements of the research objectives. CBC want to implement a repeatable data mining process which is usable in the long term to detect HMO properties. Therefore, the evaluation of the project will need to make considerations for the tools usability.

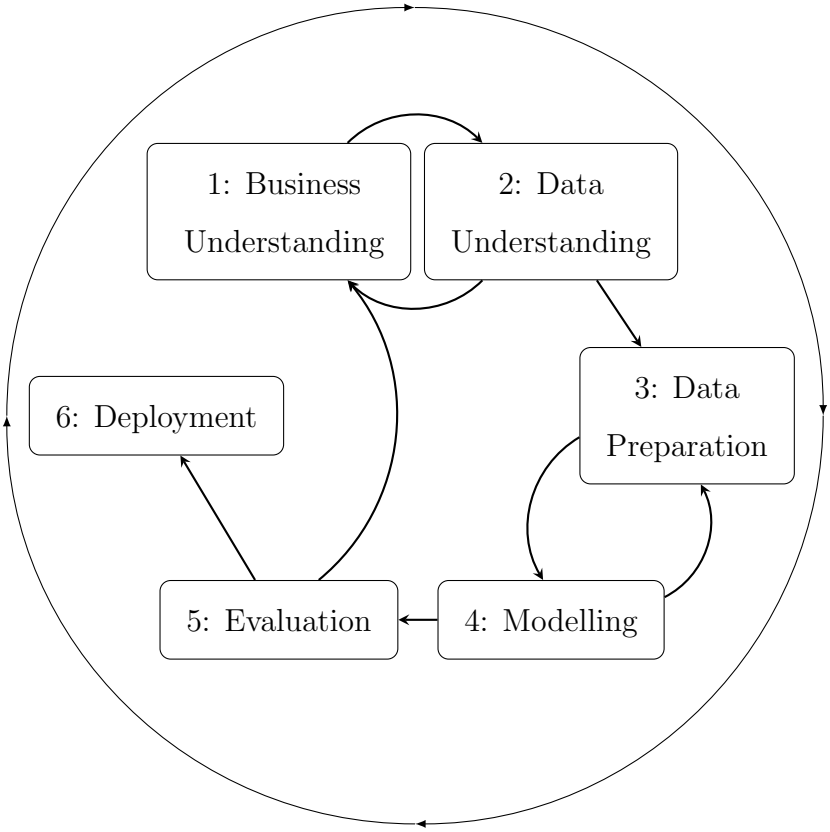


Figure 1: Pictorial representation of the CRISP-DM methodology, illustrating the relationships between each of the processes. Adapted from [1], chapter 2.

2 Theoretical Background

In section 1.2 it was established that HMO properties present themselves as a rare-event and that special techniques will be required in order to achieve good generalization performance. This section contributes to the 'Business Understanding' and 'Data Understanding' phases of the CRISP-DM methodology outlined in section 1.2. Section 2.1 documents a review of common problems associated with rare-event modelling, along with solutions which have been propounded to remedy them. Section 2.2 reviews string matching and other entity resolution techniques which are applicable to the requirements of RO1. Finally, section 2.3 closely follows the work of Steinberg et al [21] and provides a review of common classification techniques, but also evaluates them with respect to their handling of rare-events. Section 2.4 provides an overview of feature selection techniques which are required to fulfil RO2.1.

2.1 Rare Event Classification

Gary M. Weiss from AT&T Labs outlined 6 key problems which arise from the mining of rare-events in his 2004 paper "Mining with Rarity: A Unifying Framework" [22]. The 6 key problems are summarized below:

1. **Improper Evaluation Metrics**: The chosen metrics should adequately make considerations for rare classes, otherwise the data mining procedure would be unlikely to handle the rare class. For example, the performance metric 'accuracy', which computes the proportion of examples that are correctly classified by a classifier; is a flawed approach in rare-event modelling because accuracy is biased towards the majority class and is less affected by the minority class.

2. **Relative Rarity**: Relative rarity refers to a class which is rare with respect to other classes. Rules learned on rare classes are greatly impacted by relative rarity because the common classes may obscure the true effect of rare classes, which, in turn, may obscure the learned rules made.

3. **Absolute Rarity**: According to Gary Weiss, the fundamental problem with rarity is the associated lack of data, or 'absolute rarity'. Absolute rarity of a class makes it

difficult to detect regularities within the rare class because there's not enough data to model on. A class could be relatively rare if it makes up 2% of a two-class dataset, but if it was 2% of a dataset which contained 100,000,000 instances, then it could be inappropriate to refer to the class as an "absolute rare" case. Therefore, the terms "absolute" and "relative" rarity are introduced to discern between these conditions.

4. **Data Fragmentation**: Data Fragmentation is caused by data mining algorithms which employ a divide-and-conquer approach, where the original dataset is partitioned into smaller subsets, such is the case for classification and regression trees. This causes problems for rare-event classifications because commonalities in the rare cases are harder to detect when an algorithm partitions the original data into a subset which contains less data, which can induce or intensify pre-existing absolute and relative rarity problems.

5. **Inappropriate Inductive Bias**: Induction refers to generalizing from specific examples and requires extra-evidential bias. For rare cases, this bias isn't possible and induction is not possible and learning cannot occur. Many learners utilize a general bias in order to foster generalization and avoid overfitting.

6. **Noise**: By definition rare-events have fewer examples to learn from, so fewer noisy examples are required to impact the learned class.

Evaluation metrics that account for rarity prevent the prediction performance of a model from being predominated by classes which are more common. An evaluation metric should properly account for the distribution of classes in order to be dependable. In classification problems, good accuracy in classification is often a primary concern for model evaluation; however, accuracy is biased towards common classes, which makes it inappropriate for evaluating models which seek to predict rare events. Accuracy, along with a number of other performance metrics, can be derived using the confusion-matrix method (table 1). Confusion matrices contain information about actual and predicted classifications. A true negative (TN) is the number of negative tuples which were correctly labelled by the classifier, a false positive (FP) is the number of negative tuples which were incorrectly labelled by the classifier, a false positive (FP) is the number of incorrect tuples which were incorrectly labelled by

the classifier, and a true positive (TP) is the number of positive tuples which were correctly labelled by the classifier [4].

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Table 1: An example confusion matrix for a two-class classification problem [5]

The classification accuracy and error can be derived from the confusion matrix, they are given by:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}, \tag{1}$$

$$Error = \frac{FP + FN}{TN + TP + FN + FP} = 1 - Accuracy \tag{2}$$

The accuracy of a model is the fraction of predictions which were correctly labelled by the classifier, whereas the classification error is the fraction of predictions which were incorrect [5]. The recall and precision measures can also be derived using confusion matrix method. Recall is the proportion of positive tuples that detected by the classifier, while precision is the proportion of positive tuples which were correctly labelled by the classifier [4]. These measures are given by:

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Simple precision is misleading because it counts multiple predictions of the same target event multiple times. In rare event modelling, precision can be normalized to eliminate this problem by replacing the number of correct predictions with the number of target events correctly predicted [23].

Receiver operator curve (ROC) analysis and the associated area under curve (AUC) is an alternative evaluation method. AUC calculations are not biased towards one class or the

other, so the majority class does not impede the performance of the minority class. ROC graphs are two-dimensional graphs in which the true positive rate is plotted as a function of the false positive rate. It shows a trade off between selectivity and sensitivity. [24]. The AUC can be defined as

$$AUC = \int_0^1 F_0(s)dF_1(s) = \int_{-\infty}^{+\infty} F_0(s)f_1(s)ds \tag{5}$$

Where $F_0(s)$ is the true positive rate and $F_1(s)$ is the false positive rate [25]. The AUC has a statistical property of significance: the AUC of a classifier is equal to the probability learner model will rank a random positive class higher than a random negative class.

Methods of oversampling and undersampling have been used to deal with the absolute rarity, relative rarity problems and data fragmentation problems. Oversampling involves expanding instances of rare cases, while undersampling involves eliminating instances of the majority case; both methods seek to redistribute the number of rare and majority cases, so that they become comparable to one another. A common method of oversampling and undersampling is the random re-sampling technique, which consists of replicating the rare case at random until it contains as many examples as the other class, however; it has been shown that this method doesn't significantly improve minority class recognition. Furthermore, random re-sampling has been shown to increase the likelihood of over-fitting, thereby reducing model generalizability due to the creation of duplicate minority class instances (in the case of oversampling) [26, 27]. A method of oversampling which was introduced to deal with the shortcomings of the random re-sampling technique, is the "synthetic minority over-sampling technique" (SMOTE) [28]. SMOTE is an oversampling method where the rare class is over-sampled by creating "synthetic" instances. In SMOTE, the rare class is oversampled by taking each rare class sample and create algorithmically derived fake or synthetic examples based on the k minority class nearest neighbours. The synthetic examples cause the classifier to create larger and less specific set of rare classes, rather than smaller and more specific rare classes, as typically caused by the random re-sampling technique [29].

2.1.1 Active Learning

Human interaction in learning models is especially useful for rare event classification, because it can incorporate the knowledge of those who have domain expertise, aiding in the class labelling process. Active learning approaches require a user to label an example, which may be from a set of unlabelled examples or presented by the learning program based on classes which have been determined to be of low confidence and require verification from the user. The goal is to optimize the model quality by actively acquiring knowledge from human users. The user can be presented with samples to label as a means of creating training sets or estimating classification precision. A study by Pelleg & Moore [30] introduced a novel active-learning scenario in which a user works with a learning algorithm to identify useful anomalies from an astronomical survey dataset. They found that popular active learning techniques involving the random presentation of examples performed poorly when using imbalanced datasets but performed well when users were presented examples which the learning algorithm deemed low or high uncertainty.

2.2 Approximate String Matching

String matching (SM) is an important subject in the wider domain of text processing. SM consists of finding an occurrence, or all the occurrences of a string in a set of data objects and the basic precepts underpinning SM are used in practical implementations of most software paradigms [31]. SM is complicated by noisy data, which can be caused by hardware corruption, or more commonly, human error in data entry such as misspellings, typing errors or data entry without a consistent schema which is the principal issue for CBC's records of rent deposit addresses. Therefore, approximate string matching (ASM) has been introduced with the goal of finding commonalities between data objects, while allowing a limited number of errors [32].

2.2.1 Similarity Measures

One way in which ASM can be achieved is through the use of similarity measures (Table 2). One such measure is the Levenshtein distance or edit distance, which was first presented by Vladimir Levenshtein in 1966, and is defined as "the minimum number of inserts, deletions, and replacements of characters required to transform a string x into a string y" [33, 34]. The Levenshtein distance, D(x,y) for the words $x =$ "$fizzy$" and $y =$ "$dizzy$" would be D(x,y) = 1 because one substituion is required to transform s_1 into s_2. Another ASM method is the longest common subsequence (LCS), which allows operations of insertions and deletions, which have a cost of 1 [35]. It is formally described by the recurrence relation:

$$L(i,j) = \begin{cases} 0 & \text{if i = 0 or j = 0} \\ 1 + L(i-1, j-1), & \text{if } x_i = y_i \\ max\{L(i-1,j), L(i,j-1)\}, & \text{if } x_i \neq y_i \end{cases} \tag{6}$$

Where $L(i,j)$ is the length of the LCS between strings x[1:i] and y[1:j] where (i,j) represents the string index. The LCS of a string can be used to compute a corresponding Levenshtein distance. For two strings of length m and n the LCS is given by

$$L = \frac{(m + n - D(i,j))}{2} \tag{7}$$

Where $D(i, j)$ is the Levenshtein distance. $D(i, j)$ can be computed for a pair of strings (σ_1, σ_2) using the dynamic equation:

$$D(i,j) = \begin{cases} 0, & \text{if i = 0 and j = 0} \\ D(0, j-1+\delta(\lambda, y_j)), & \text{if i = 0 and j > 0} \\ D(i-1,0)+\delta(x_i,\lambda), & \text{if i > 0 and j = 0} \\ min \begin{cases} D(i,j-1)+\delta(\lambda,y_j), \\ D(i-1,j)+\delta(x_i,\lambda), \\ D(i-1,j-1)+\delta(x_i,y_j) \end{cases} & , \quad \text{if } x_i \neq y_j \end{cases} \tag{8}$$

Where $\delta(\lambda, \sigma)$, $\delta(\sigma, \lambda)$ and $\delta(\sigma_1, \sigma_2)$ represent the cost of insertion, deletion and substitution, respectively.

Another ASM method which can be used is the Hamming distance. The traditional definition of a Hamming distance is: "given two strings of the same dimension, the Hamming distance is the minimum number of edits needed to change one string into the other" [36]. The Hamming distance requires that the strings are of the same length, so would be a disadvantaged approach to take when matching property addresses from data with an inconsistent schema. The Levenshtein distance allows for insertions, so works well with data which may contain typing errors such as unnecessary characters.

Similarity Measure	**Summary**
Levenshtein Distance	Allows operations of insertions, deletions and replacements; where all operations cost 1.
Longest Common Subsequence	Allows operations of insertions and deletions, which have a cost of 1.
Hamming Distance	Allows only operations which conduct replacements, which have a cost of 1.

Table 2: A summary of common similarity measures used in approximate string matching, using simplified definitions.

2.2.2 Hierarchical Clustering

Clustering analysis is a common approach in document duplication detection, which is used to identify records that potentially refer to the same entity [37]. Clustering algorithms seek to group a set of data objects into subsets or clusters which are coherent internally, but clearly discernible from each other. Hierarchical clustering is the decomposition of a given set of data objects. Hierarchical clustering can either be agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed. Agglomerative clustering algorithms commence by designating each data object as a singleton cluster. The clusters are successively agglomerated until a stopping criterion is met or all data objects have been merged into a single cluster. Conversely, divisive clustering algorithms begin by designating a series of data objects as a single cluster and then recursively partitioning clusters until singleton clusters are reached [38]. A tree structure called a dendrogram can be used to visualize the process of hierarchical clustering (Figure). In a dendrogram, merges are represented by horizontal lines and the distance between merges are represented by vertical lines.

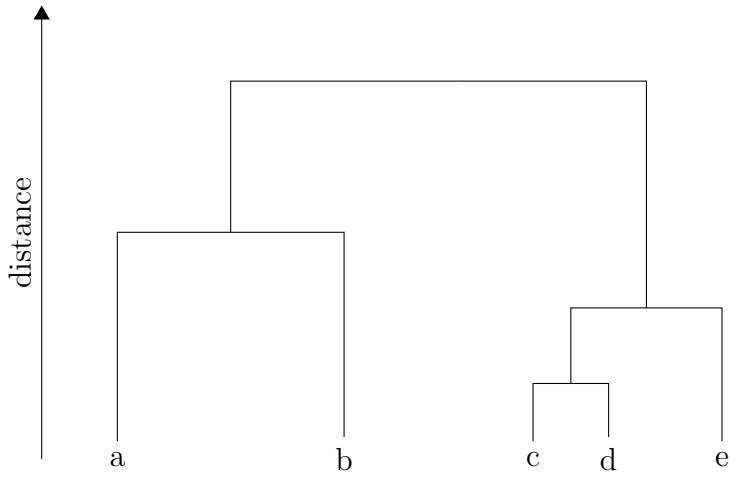


Figure 2: Dendogram illustration of divisive hierarchical clustering. Horizontal lines represent a merge, while vertical lines represent distance.

To compensate for the rigidity of a merge or split, the quality of hierarchical agglomeration can be improved by analysing object linkages at each hierarchical partitioning [39].A splitting or merging criterion can be established by distance or similarity metrics. Given a pair of clusters (C_i , C_j) containing points x_i and x_j where $x_i \in C_i$ and $x_j \in C_j$; the maximum distance between the clusters is given by:

$$d_{max}(C_i, C_j) = \max_{x_i \in C_i, x_j \in C_j} |x_i - x_j| \tag{9}$$

A splitting or merging criterion which uses the maximum distance metric is known as single-linkage clustering. Complete-linkage clustering uses the minimum distance metric, the minimum distance between a pair of clusters is given by:

$$d_{min}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} |x_i - x_j| \tag{10}$$

In centroid clustering, splitting and merging criteria are based on the distance between a pair of cluster's centroids i.e. the average inter-cluster distance. The average distance between points within the clusters is given by:

$$d_{Av}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x_i \in C_i, x_j \in C_j} |x_i - x_j| \tag{11}$$

The similarity of two objects is given by the inverse of the distance. Hierarchical agglomerative clustering (HAC) techniques are variations of a single approach: starting with clusters

which contain a single object and successively merging clusters until one remains [40]. Algorithm 1 shows a basic HAC algorithm.

Algorithm 1 Hierarchical Agglomerative Clustering (HAC) [40]

Input A set of data objects (x_i,...,x_N)

Output A set of clustered data objects ($C_i, ..., C_N$)

1: **procedure** HAC(x_i,...,x_N)
2: **for** $n \leftarrow 1$ *to* N
3: **do for** $i \leftarrow 1$ *to* N
4: *Calculate* $d(C_i, C_j)$
5: **repeat**
6: *Merge the two closest clusters*
7: **until** *one cluster remains*

2.3 Classification Techniques

Classification has been defined as models which predict categorical (discrete, unordered) class labels [41]. In machine learning algorithms, if instances are given with known labels then this is known as supervised learning, conversely; if instances are given with unknown labels then this is known as unsupervised learning [42]. In this project supervised learning is adopted because data collection makes use of instances with known class labels (RO2). Furthermore, the objective is to develop a classifier which predicts a known class label, rather than discovering new class labels. In this section, a concise, rather than comprehensive review of common classification techniques based on research by Steinberg et al is presented and then further evaluated with respect to their handling of rare-events [21].

2.3.1 Classification by Backpropagation & Neural Networks

Classification by backpropagation is a popular type of neural network (NN) algorithm, which performs learning on a multilayer feed-forward NN. A feed-forward NN only allows signals to transfer from inputs to outputs. A NN is a series of interconnected input and output nodes in which each connection has a weight associated with it. Backpropagation learns by performing an iteration process on a data set of training tuples and comparing the network's predictive performance for each tuple with the respect to the actual known target value. In the learning phase a NN learns by adjusting the weightings of input tuples to optimize class label predictability [2, 43]. Modifying the weights of the input connections leading to the hidden units is based on the effect of the unit on the predictive performance of the model. Modification is usually achieved through a mathematical optimization technique known as gradient descent [43]. In backpropagation each unit in the hidden and output layers an input and then applies an activation function to it, most commonly, the sigmoid function is used:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

With its derivative:

$$\frac{df(x)}{dx} = f(x)(1 - f(x)) \tag{13}$$

The sigmoid function maps input domains onto the smaller range of 0 to 1. The sigmoid function is nonlinear and differentiable, allowing the backpropagation algorithm to model

classification problems that are linearly inseparable [2].

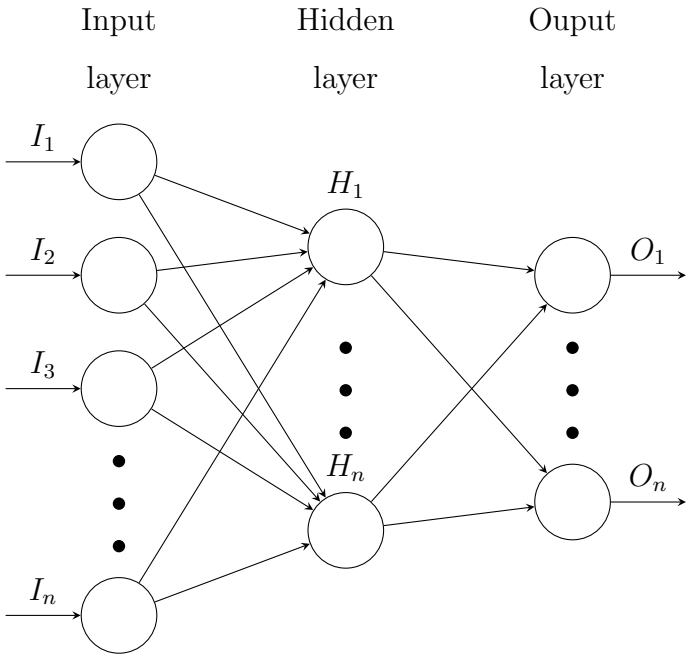


Figure 3: Pictorial representation of a multilayer feed-forward neural network. A multilayer feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer [2, 3].

Neural network techniques offer a number of advantages, including the ability to implicitly detect complex non-linear relationships between dependent and independent variables [44]. NN techniques are noted in literature for having high robustness and there is limited performance degradation in the presence of increasing amounts of noise [45]. Conversely, NN techniques involve long training times with a high number of weights and have been noted as having a proneness to overfitting [44, 45]. Furthermore, they are difficult to interpret when compared to other methods such as decision trees, which made them less desirable for data mining procedures in the past [2]. In rare-event classification, a significant issue for NNs comes from data imbalance. For example during a training phase, the classifier estimates class presence probabilities and calculates an error based on a loss function. In rare-events, the target class in not present in a significant proportion, so the classifier will be biased [46]. Choe et al investigated a way to remedy this and found that non-representative sample stratification schemes made rare-event classes have a better chance of being included in the

sample used for training NNs and improved classification accuracy [47]. Some NN algorithms that have been propounded to account for rare-events includes fuzzy ARTMAP, which can autonomously learn, recognize, and make predictions about rare events [48]. Further research involving rare-events includes rare sound event detection system using combination of 1D convolutional NN and a recurrent NN [49].

2.3.2 Classification and Regression Trees

Classification and Regression Trees (CART) are a binary recursive partitioning procedure capable of processing continuous and nominal attributes as targets and predictors [21]. They are a type of decision tree, which is a classification method used for approximating discrete-valued target functions, in which the learned function is represented by a flowchart like structure (Figure) [50]. In CART splitting, an instance which meets the splitting criterion is branched left and instances which do not meet the splitting criterion are branched right. At a node t, the best split s is chosen to maximise the a splitting criterion Δi(s, t). When the impurity measure for a node can be defined, the splitting criterion corresponds to a decrease in impurity. The Gini impurity at a node t is defined as

$$i(t) = \Sigma_{i,j} C(i \mid j) p(i \mid t) p(j \mid t) \tag{14}$$

Where $C(i \mid j)$ is the cost of misclassifying a class j case as a class i case, $p(i \mid t)$ is the probability of case i at node t and $p(j \mid t)$ is the probability of case j at node t[1]. When cost-sensitive learning is not used $C(i \mid j) = 1$. The Gini splitting criterion is the decrease of impurity defined as

$$\Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R) \tag{15}$$

Where p_L and p_R are the respective probabilities of a case being branched to the left daughter node t_L or right daughter node t_R [51].

Simple CART models struggle with imbalanced data because with each partition, there are fewer examples of the rare case to learn from, which intensifies issues such as relative

[1]Taken from a previous project of mine: "ASDM Assignment: Data Mining using SAS and R", submitted to the University of Salford.

and absolute rarity, however; this can be remedied with oversampling and undersampling techniques . Furthermore, methods unique to decision trees can be used to improve predictive accuracy, such as pre-pruning and post-pruning. In pre-pruning the tree is pruned by halting construction of the tree when a condition is met and in post-pruning a full tree is constructed, then retrospectively pruned based on analysis of the fully constructed tree [4]. CART models can be adapted to effectively deal with rare events. For example, a study by J. Herbert found that tree-based methods performed well with rare events when detecting factory production line failures. Herbert also concluded that tree-based classifiers can detect relationships that are not readily detectable with linear regression modelling such as logistic regression, but regression modelling was better suited to discovering primary relationships between variables [52].

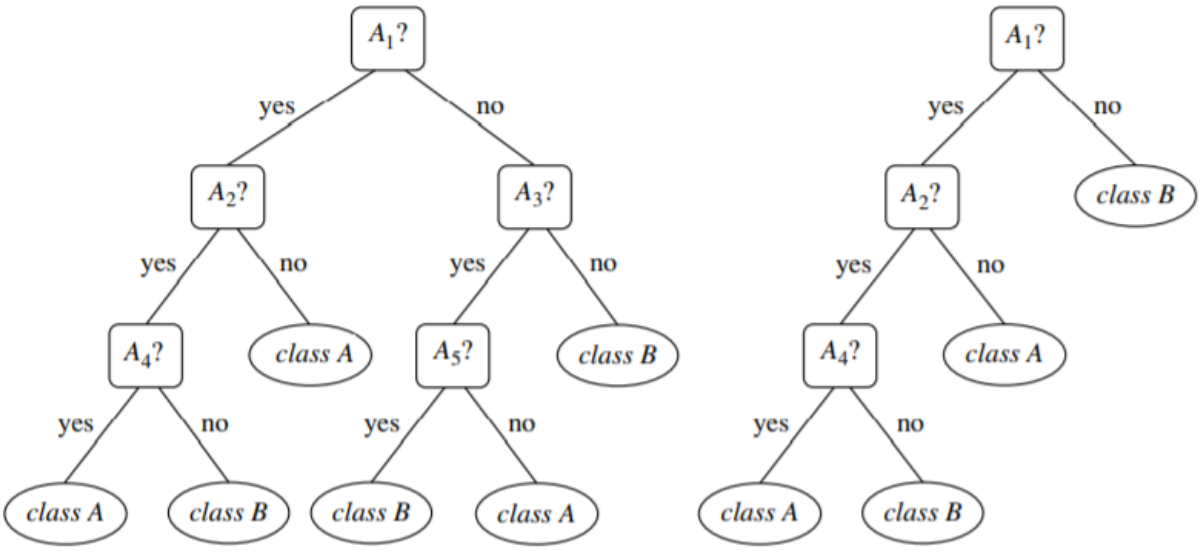


Figure 4: CART flowchart for a binary classification problem. A_x represents the partition condition and class A/B represents the class label. The left hand side shows a pre-pruned tree, whereas the right hand side shows a post-pruned tree [4].

2.3.3 Logistic Regression

In logistic regression, a binary categorical dependent variable Y is regressed against a set of independent variables, $X = (X_1, X_2, ...X_N)$ and a set of population regression coefficients β, which are estimated from data, denoted by:

$$B_g = \begin{pmatrix} \beta_{g1} \\ \vdots \\ \beta_{gp} \end{pmatrix} \tag{16}$$

The objective of logistic regression is to find a model with predictive capability of the form:

$$\log_e \left(\frac{\rho_g}{\rho_1} \right) = \log_e \left(\frac{P_g}{P_1} \right) + \beta_{g1} X_1 + \beta_{g2} X_2 + \cdots + \beta_{gp} X_\rho \tag{17}$$

Where ρ_g = P(Y=g | X), i.e. the probability of Y being one of two outcomes for a given independent variable. The quantities P_1, P_g represent prior probabilities of outcome membership [53].

Logistic regression is a popular classification technique which is well documented in literature in the form of implementations by political scientists to model rare events such as wars, vetoes and political activism. Logistic regression is noted to perform poorly with small-sample biases, and this bias inversely correlates heavily with the number of rare cases in the sample [54]. One method to deal with rare cases in logistic regression modelling is known as the 'Firth method', which is a general approach to reducing small-sample bias in maximum likelihood estimation [55]. Logistic regression employing Firth-type penalisation is a popular method to reduce the relative rarity bias of β regression coefficients, however; reducing the bias in the estimates of β coefficients comes at the cost of introducing bias in the predicted probabilities [56] .

2.3.4 Naïve Bayes' Classifiers

Naïve Bayes' Classifiers (NBC) are statistical classifiers which are derived from Bayes' theorem:

$$P(i|j) = \frac{P(j|i)P(i)}{P(j)} \tag{18}$$

A NBC discerns between a class variable C_k and a set of predictor variables. Furthermore, NBCs measure the dependence of a class variable on each predictor separately, and models the independence between other predictors [57], which can be represented by a probability distribution given by:

$$p(i|j) = p(C_k) \prod_{i=1}^{N} p(x_i|C_k) \tag{19}$$

Where $p(x_i|C_k)$ is the parameter probability of a predictor x_i when the class C is true. The parameter probability is dependent on the number of instances of x_i and tends towards 0 for rare features, which makes NBCs unsuitable for data mining projects which seek to model rare features [58].

2.4 Feature Selection

Feature selection methods aim to reduce model dimensionality by reducing the number of irrelevant features in a model, thereby increasing the model's generalizability and predictive accuracy [59]. Irrelevant features are defined as features which do not affect a target outcome in a statistically significant way. In real-world two-class classification problems relevant features are often unknown, and candidate features are usually implemented into prototype models, which can diminish a model's predictive accuracy; this is known as the ad-hoc approach, and usually requires substantial domain expertise. A more objective approach to feature selection comes in the form of filter, wrapping and embedding techniques. Filters select variables by ranking them according to correlation coefficients or testing them against some criterion outside of a predictive model. Wrapper methods assess subsets of variables according to their usefulness to a given predictor. Embedded filtering makes use of both of the aforementioned methods, whilst simultaneously performing classification, such is the case for some decision trees, which may partition data based on a filtering criterion. Feature selection facilitates data understanding and remedies the so called "curse of dimensionality" to improve prediction performance [60].

One filter method comes in the form of the Pearson correlation, which is a measure of linear dependence between two variables, which yields a value between -1 and 1. A Pearson correlation of 1 indicates a perfect positive linear relationship, while a correlation of 0 indicates no relationship. For a feature with values x and classes y, the Pearson correlation coefficient is given by:

$$\hat{\rho}_{x,y} = \frac{\sum_{i=1}(x_i - \bar{x}_n)(y_i - \bar{y}_n)}{\sqrt{\sum_{i=1}(x_i - \bar{x}_n)^2}\sqrt{\sum_{i=1}(y_i - \bar{y}_n)^2}} \tag{20}$$

Where $\bar{x}_n$ is the mean value of x and $\bar{y}_n$ is the mean value of y. [61]. The Pearson correlation is applicable to binary, continuous, or single value target variables. Irrelevant features should have a near-zero Pearson correlation, however; the Pearson correlation is limited by the fact that it does not capture non-linear relationships between variables. Therefore, a non-linear relationship may exist and features which have a near-zero Pearson correlation could still be relevant [62]. An alternative method of measuring the relationship between two variables is

the Chi-square (χ^2) test, which measures the lack of independence between a term x and a category y. The χ^2 statistic can be defined as

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \tag{21}$$

Where o_{ij} is the observed frequency of the joint event and e_{ij} is the expected or theoretical frequency of (x_i, y_i) [63]. The χ^2 statistic has a natural value of 0 if (x_i, y_i) are independent [64]. Features can also be selected based on information gain (IG). IG is a filter method which measures the reduction of entropy, irrelevant features would be expected to give no IG. It is also a common technique for embedded feature selection in classification trees where partitions can be made based on maximising the reduction in entropy. Entropy is mathematical defined as

$$E(x) = -\sum_{i=1}^{N} p_i log_2(p_i) \tag{22}$$

Where p_i is the probability of an instance, x belonging to a class i, $\frac{|x_i|}{x}$. IG a measure of the reduction in Entropy, ΔE(x):

$$\Delta E(x) = IG(x) = E(x) - \sum_{j=1}^{N} \frac{|x_{i,j}|}{x} \cdot E(x_j) \tag{23}$$

Where E(x) is the root entropy, E(x_j) is the entropy at the j^{th} partitioning subset and $\frac{|x_{i,j}|}{x}$ denotes the probability of an instance x belonging to a class i at the j^{th} partition [65].

3 Specification and Design

This section details the requirements needed to achieve the research objectives and outlines the methodological solution proposed at a high level.

3.1 Address Clustering

The data flow diagram for the address clustering methodology is shown in Figure 5. The data normalization node is required because rent deposit data held by the council comes in the form of semi-structured text data with an inconsistent schema, which designate property addresses. Therefore, measures are needed to normalize the data, so that appropriate analysis can be done. The sampling node is required because the number of possible unique pairs of addresses in council tax records, ${}^{n}P_{r}$, is given by:

$$ {}^{7386}P_2 = \left(\frac{7,386!}{(7386-2)!} \right) \div 2 = 27,272,805 \tag{24} $$

Therefore, sampling the data makes the solution more scalable by requiring less computational power for larger datasets and in turn, will improve the potential long-term usability for the tool. It is easy for humans to classify whether a pair of addresses refer to the same property or if they're a potential HMO, especially for Housing Officers who frequently work with HMO properties and have become accustomed to the minutiae associated with HMO addresses; however, labelling 27,272,805 pairs of addresses every month is not a feasible task. The active learning instance is used to label paired addresses for training and validation sets. As mentioned in section 2.1.1, popular active learning techniques involving the random presentation of examples perform poorly when using imbalanced datasets but performed well when users were presented examples which were deemed low or high uncertainty. The objective of this phase in the methodology is to present the user with pairs of addresses of high or low similarity based on the similarity calculations done in the previous phase, so that the most impactful information can be found. The final phase is to cluster addresses together based on how similar they are to each other.

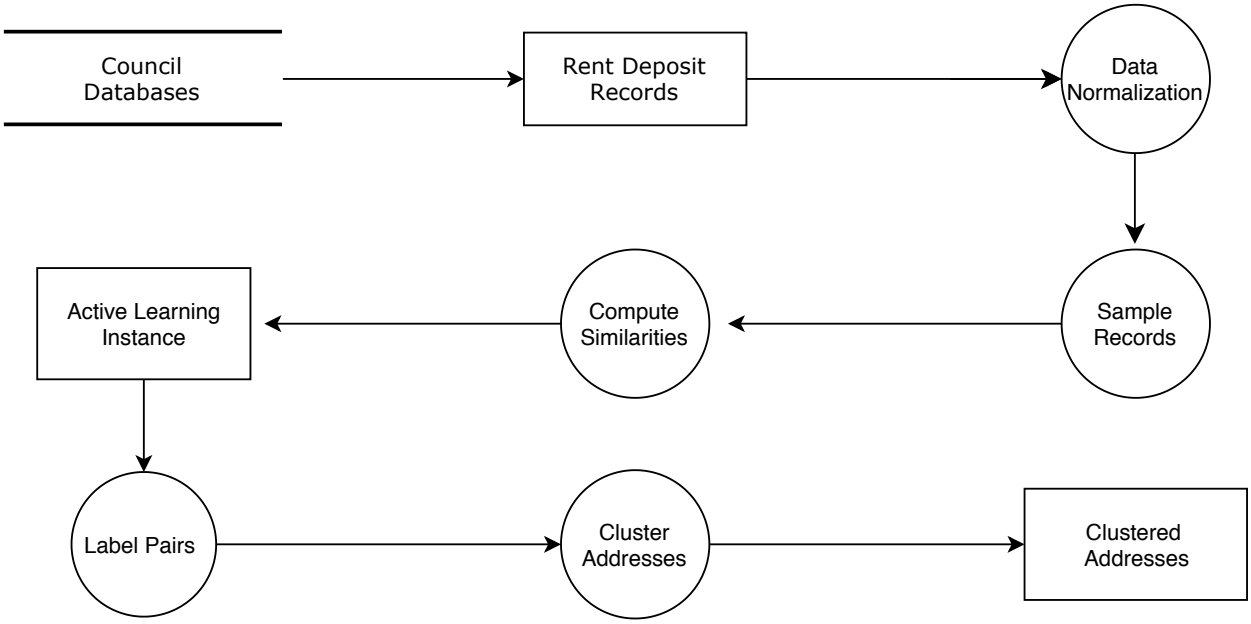


Figure 5: Data flow diagram showing the address clustering process design at a high level.

3.2 Property Advertisement Classification

The first step in the methodology is data collection. A HMO property by definition is occupied by persons living in two or more separate households totalling 5 or more persons. Local authorities have the ability to license landlords operating a HMO which has less than 5 occupants. CBC are also interested in properties with 3 or more people living in them, forming more than 2 households. Therefore, it is a requirement is made to only collect data on properties with at least 3 bedrooms. The second step is to establish a series of candidate features. In this case candidates were selection based on their ease of access in property advertisements. In order to be usable in the real word, the candidate features must be accessible from the advertisements. The data cleansing and feature selection phases are used to prepare the data for modelling. Feature selection aims to reduce model dimensionality by reducing the number of irrelevant features in a model, thereby increasing the model's generalizability and predictive accuracy. Irrelevant features are defined as features which do not affect a target outcome in a statistically significant way.

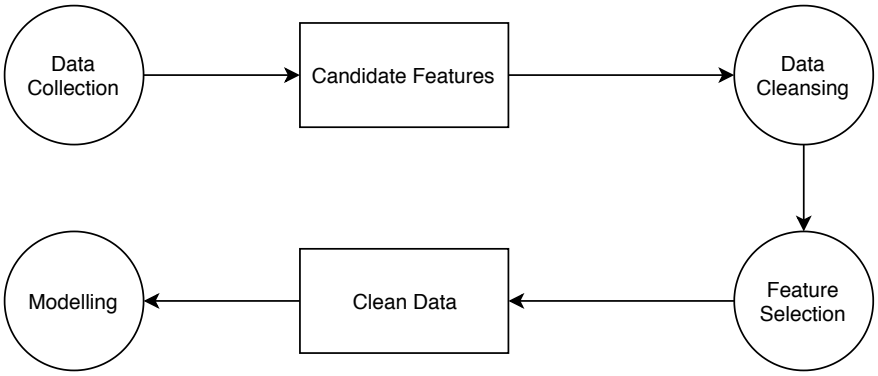


Figure 6: Data flow diagram for property advertisement classification

4 Development and Implementation

This section contributes to the 'Data Preparation' and 'Modelling' phases of the CRISP-DM methodology. Section 3.1 contains the methodology used to achieve research objective 1 and reports how entity resolution was performed on rent deposits records to detect potential HMO properties. In section 3.2, the methodology used to achieve research objective 2 and its sub-objectives is reported. Property advertisement data was collected and modelled using recursive partitioning to classify property advertisements as HMO and non-HMO types as an attempt to detect HMOs without relying on council data.

4.1 Address Clustering

In the introduction section of this report it was explained that according to the 2004 Housing Act, HMOs which require mandatory licensing requires a dwelling which is occupied by five or more person. Therefore, properties which were detected to have more than 5 rent deposit records were designated as mandatory-type HMO properties. In coordination with CBC, a provision was also made to designate properties containing 3 or 4 rent deposit records as non-mandatory HMO properties because local authorities can choose to license landlords operating a HMO that is too small to require mandatory licensing. CBC would like to monitor these types of properties, but would still like to clearly discern between mandatory and non-mandatory types.

4.1.1 Rent Deposit Datasets

The datasets were provided by CBC and have briefly been described below. Rental deposit datasets may be used by a local authority for the purposes connected with the exercise of the authority's functions under any of Parts 1 to 4 of the Housing Act 2004 in relation to investigating whether an offence has been committed under any of those parts. The datasets are updated every month. The information can be supplied to a person outside of the local authority who is providing a service regarding the purpose previously stated and is only to be held by a local authority for the least amount of time required[2].

[2]This notice came with the datasets

Deposit Protection Service (DPS) Dataset: 4,994 instances of 5 attributes, described privately rented property, landlord and managing agent addresses. This dataset does not contain explicit information regarding the number of tenants living at the properties.

MyDeposits Dataset: 902 instances of 20 attributes, describing privately rented property addresses, landlord or managing agent addresses. This dataset does not contain explicit information regarding the number of tenants living at the properties. The postcodes presented in this dataset were obtained from www.doogal.co.uk in 2016. Any subsequent changes after 2015 are not reflected in the dataset.

Tenancy Deposit Scheme (TDS) Dataset: 1,490 instances of 14 attributes, describing privately rented property addresses, landlord or managing agent addresses and the number of tenants at the privately rented property. The landlord address as provided by the letting agent are not mandatory for the purposes of deposit protection. Where they have not been provided the landlord address will has been shown as "Landlord address not provided by letting agent".

4.1.2 Python Implementation

The TDS dataset contained explicit information regarding the number of tenants living at the properties, so a simple measure was taken to filter the TDS dataset to return properties containing 3 or more tenants. Shown below:

```
1 import pandas as pd
2 df = pd.read_csv("D:\MSc_Project\Data\TDS_Deposits\TDS.csv",header=3,
     index_col=False) #load TDS data
3 df1 = pd.DataFrame(df)
4 df2 = df1[df1['Number_Of_Tenants'] >= 3] # filter for  >3 tenants
5 TDS_Clean = df1[~df1['Number_Of_Tenants'].isin(df2) == False ]
6 df_sum = df1.groupby('Number_Of_Tenants')\
7         ['TDS_Reference_Number'].count()\
8         .reset_index(name="count")
```

MyDeposits and DPS data did contain explicit information regarding the number of tenants living at properties and also lacked a consistent schema. For example, initial inspection of the data showed that some addresses contained obvious spelling mistakes, unnecessary characters and inconsistent use of capitalization, so more sophisticated measures were required to find properties with 3 or more tenants in the MyDeposits and TDS datasets. A data reduction measure was taken to remove properties with postcodes which didn't appear 3 or more times in the dataset. An attempt to normalize address lines was made, which included removing line breaks, unnecessary spaces and punctuation such as quotation marks, commas and colons. The function is shown below:

```
import re
def NormalizeAddress(addressline):
        addressline = re.sub(',','', addressline)
        addressline = re.sub('/','', addressline)
        addressline = re.sub('[^A-Za-z0-9]+', '', addressline)
        addressline = re.sub('[','', addressline)
        addressline = re.sub(']','', addressline)
        addressline = re.sub('\[[^]]*\]', '', addressline)
        addressline = re.sub(r'[?|$|.|!]',r'',addressline)
        addressline = re.sub('␣␣+', '', addressline)
        addressline = re.sub('\n', '', addressline)
        addressline = addressline.strip().strip('"').strip("'")\
        |       .lower().strip()
        if not addressline:
        |       addressline = None
return addressline
```

The 'dedupe' Python library was used to match addresses which referred to the same property. Dedupe was created by Forest Gregg and Derek Eder, it uses machine learning techniques such as hierarchical clustering to perform de-duplication and entity resolution [66, 67]. In this case, it was used for entity resolution on property addresses. The datasets

combined contained 7,386 records of rent deposit addresses. The number of possible unique pairs of addresses, ${}^{n}P_{r}$, is given by:

$$ {}^{7386}P_{2} = \left(\frac{7,386!}{(7386-2)!} \right) \div 2 = 27,272,805 \tag{25} $$

Therefore, the 'sample' function was used to specify a subset of records from the datasets to train on. It takes a mixture randomly selected pairs of addresses which are more likely to be duplicates based on Levenshtein distance. This makes the solution more scalable by requiring less computational power for larger datasets and in turn, will improve the potential long-term usability for the tool. The sample function has the general structure:

```
sample(data[, [sample_size=15000[, blocked_proportion=0.5[,
   original_length]]])
```

Where 'data' (dict) refers to the user-defined dictionary of records, 'sample_size' (int) is the number of record tuples to return, 'blocked_proportion' (float) is the proportion of record pairs to be sampled from similar records and 'original_length' denotes the size of the parent dataset. In this experiment the 'sample' function retained its default settings. An active learning approach was taken to label training data and validation data (2.1.1). It is easy for humans to match address records but it is not feasible or practical for an individual or group of people to label 27,272,805 pairs of addresses every month, which is why a machine learning approach was taken. Additionally, the active learning instance can ask the user to label instances which have low confidence, increasing the learning capability of the method. The 'consoleLabel()' function was used to start an active labelling instance from the command line (Figure 2), using training data defined by 'sample()'.

```
starting active labeling...
Property Address : 83 west dock, the wharf, leighton buzzard, bedfordshire
Property Postcode : lu7 2aj

Property Address : 79 east dock, the wharf, leighton buzzard, bedfordshire
Property Postcode : lu7 2la

0/10 positive, 0/10 negative
Do these records refer to the same thing?
(y)es / (n)o / (u)nsure / (f)inished
```

Figure 7: Dedupe active labelling instance generated using the 'consoleLabel()' function. It presents a mixture of random, dissimilar and similar pairs for the user to label.

Address properties were labelled until the minimum requirement was made for a deduper training set, at least 10 positive and 10 negative examples. The 'threshold' function was used find the optimal trade-off between precision and recall. It has the general structure:

```
threshold(data[[, recall_weight=1.5]])
```

Where data (dict) is a dictionary of records and 'recall_weight' (float) sets the tradeoff between precision and recall. Threshold recall weighting was increased incrementally from 0 in units of 0.2 until the recall maximized. The precision and recall was measured for each increment of the threshold recall weighting and a P-R plot was made, so that the optimal trade-off between precision and recall could be interpreted. The 'train()' function was used to learn pairwise classifier and blocking rules from the labelled training data. Blocking rules refer to features which duplicate pairs have in common. The 'train()' function has the general structure:

```
train([recall=0.95[, index_predicates=True ] ])
```

Where 'recall' (float) is the proportion positive class pairs in the training set that should be used and 'index_predicates' (bool) is used to specify whether predicates that rely upon indexing the data should be used. In this experiment recall was set to 0.5, which oversamples the rare case such that the proportion of rare cases contributes to 50% of the training set.

The 'match' function identifies records that all refer to the same entity using hierarchical clustering with centroid linkage. It returns tuples containing a sequence of record ids and has the general structure:

```
match(data[, threshold = 0.5[, generator=False]]])
```

Where 'data' refers to the user-defined dictionary of records, 'threshold' is a number between 0 and 1 which refers to the minimum probability required for a matching record to be returned. 'Generator' is a boolean parameter which when set to true, will return a sequence of clusters instead of a list. Match retained its default settings in this experiment. Once the clustering process took place. A Python function from the dedupe documentation was used to write the original data back into a CSV with a new column "Cluster ID" which indicated which records referred to each other. The new CSV file was then split into two CSVs, one for mandatory type HMOs and one for non-mandatory type HMOs. The same process was repeated except with the purpose of finding the total number of properties represented by the datasets, so that consistency checks could be carried out when evaluating the results. Preliminary testing found that there was little variation between precision and recall when altering the threshold weighting for DPS data. Therefore, the threshold weighting retained its default setting and P-R curve was not made for the DPS data when determining the total number of properties.

4.2 Advertisement Classification

4.2.1 Data Collection

44 property listing on Rightmove were reviewed. 6 candidate predictors were introduced based on their ease of access on property the advertisements (Table 3). Information for 44 property advertisements was recorded. 22 of the property listing were HMO properties and the remaining were non-HMOs. The minimum number of bedrooms was 3 because a HMO by definition requires at least 3 residents.

Variable	Type	Description
HMO (Target)	Binary Categorical	Designates HMO status. 'Y' for HMO, 'N' for non-HMO
Number of Bedrooms	Integer	The number of bedrooms in the property listing.
EPC Rating	Categorical	The energy efficiency ratings: A, B, C, D, E, F, G or Undeclared
Bills	Categorical	Inclusive, Non-Inclusive or Partial
Rent	Numeric	The cost of rent per month (£). Ranges from 210 to 565
Deposit	Numeric	Indicates value of deposit for property (£)
Furnishing	Categorical	'F', 'U', 'P' for furnished, unfurnished and partially furnished

Table 3: Table showing candidate variables in training model.

4.2.2 Selection of Classification Technique

When comparing variables of different types, it is not possible to perform regression or develop a linear model of them. This is illustrated in the scatter plot (Figure), it shows that modelling the predictors in linear fashion doesn't provide meaningful information to deduce. A CART model was chosen for the classification model. The CART model performs a method of embedded feature selection when partitioning data, which is an appropriate solution to the aforementioned issue. NNs were another possibility, but due to the limited data, NNs were disregarded because of their proneness to overfitting which is documented in literature. Furthermore, interoperability is also another consideration made. CART models are easier to interpret when compared to NNs, so this wouldn't be appropriate avenue when presenting research to Housing Officers who are not trained in their use.

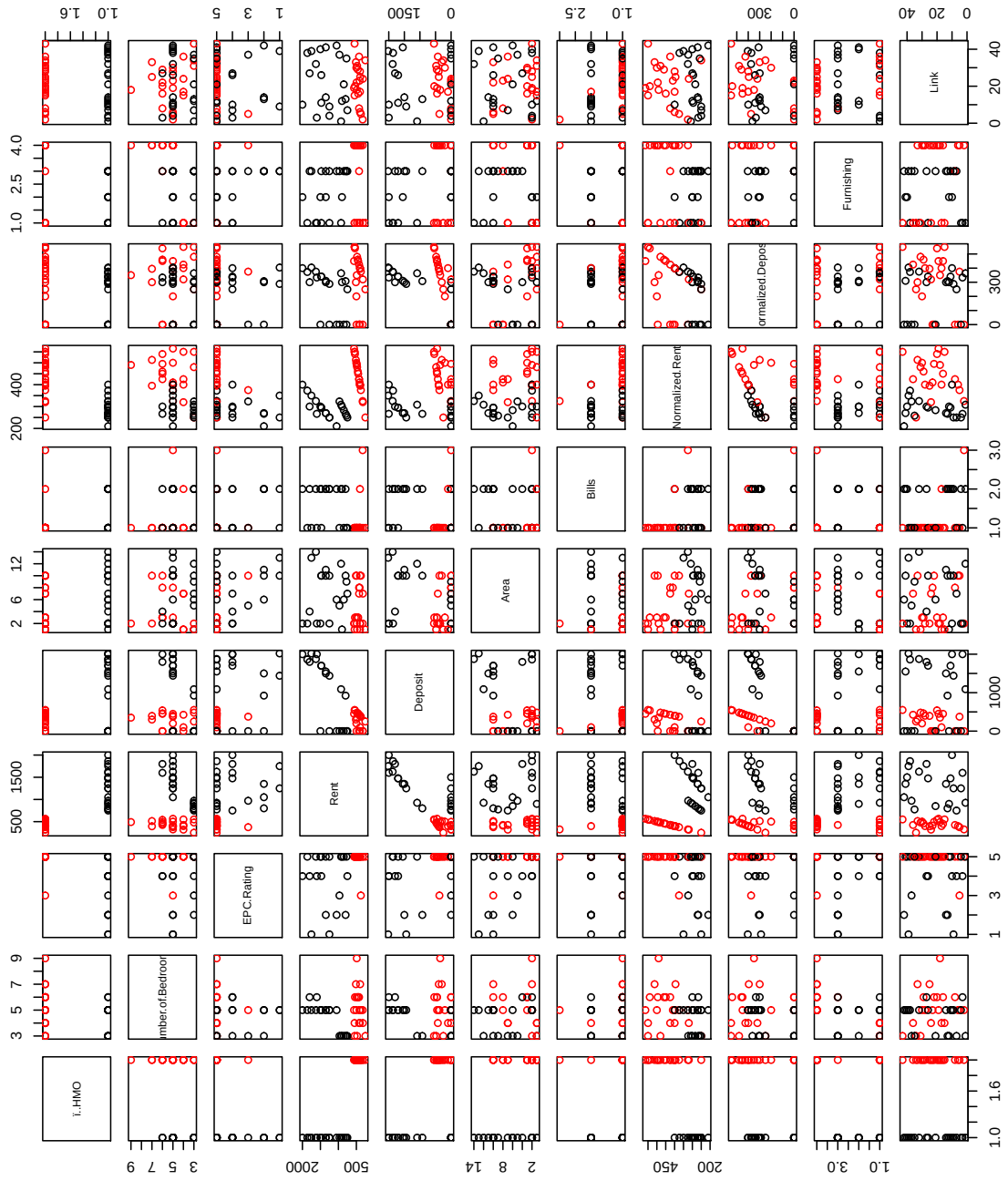


Figure 8: Scatterplot matrix of candidate features.

4.2.3 R Implementation

The property advertisement dataset was loaded into RStudio. The data was partitioned with a ratio of 80:20 into training and validations sets, respectively. The 'rpart' package was used to create a decision tree which follows the procedure outlined in section 2.3.2 [68]. The general command structure of 'rpart' is:

```
rpart(formula, data, weights, subset, na.action = na.rpart,
   method,
model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)
```

In this experiment the arguments (weights, subset etc.) assumed their default settings except 'data' and 'method'. Data was set to the partitioned training set (designated as train) and the method was set to class. The 'rpart.plot' package was used for visualization of the tree. The plotting command 'rpart.plot' has the general structure:

```
rpart.plot(x = stop("no␣'x'␣arg"),
type = 2, extra = "auto",
under = FALSE, fallen.leaves = TRUE,
digits = 2, varlen = 0, faclen = 0, roundint = TRUE,
cex = NULL, tweak = 1,
clip.facs = FALSE, clip.right.labs = TRUE,
snip = FALSE, box.palette = "auto", shadow.col = 0, ...)
```

Each node displays the classification, the probability of each class at that node and the percentage of the total instances contained at that node. The DT model built from the training data was tested against the the validation data using the 'predict' command. The predict command has the general structure

```
predict (object, ...)
```

The command structure was customized to classify the target of the validation data. The resultant command takes the form:

```
predict(my_tree, newdata=validate, type="class"), validate$HMO)
```

4.3 Performance Metrics

The performance of both research objectives was evaluated using the confusion-matrix based method outlined in section 2.1 and shown again here (Table 4). As stated earlier: in classification problems, good accuracy in classification is often a primary concern for model evaluation; however, accuracy is biased towards common classes, which makes it inappropriate for evaluating models which seek to predict rare events. Therefore, the performance of the address clustering tool was principally evaluated by its classification precision. The precision was estimated using the labelled examples in the active learning phase. This project made use of the targeted precision metioned in section 2.1, which is defined as the number of target events which were correctly predicted [23].

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Table 4: An example confusion matrix for a two-class classification problem [5]

Accuracy was applicable to the decision tree model because the rare class represented a larger proportion of the dataset, preventing bias in accuracy. The classification accuracy and error can be derived from the confusion matrix, they are given by:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}, \tag{26}$$

$$Error = \frac{FP + FN}{TN + TP + FN + FP} = 1 - Accuracy \tag{27}$$

5 Results

5.1 Address Clustering Results

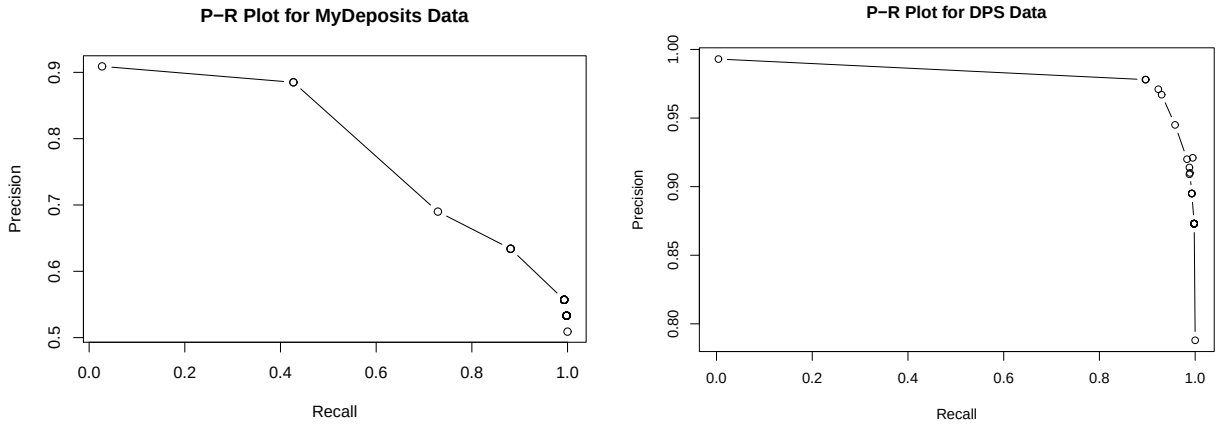
Filtering of the TDS dataset found 47 properties with 3 or 4 tenants and 1 property with 5 or more tenants. The results of the address clustering tool are shown in table 4. Figure 7 shows the P-R curves for the clustering tool, which were used to manually interpret the optimal P-R tradeoff. A total of 5,357 properties were found in the rent deposit records (Table 5).

Dataset	No of Clusters	Precision (%)	Recall (%)	No of HMOs	No of Mandatory HMOs
MyDeposits	7	86.8	44.5	1	1
DPS	160	97.1	91.3	16	3
TDS	NA	NA	NA	47	1

Table 5: Table showing information regarding address clusters for MyDeposits and DPS data. The No of HMOs column refers to the number of properties with more than 3 deposits and the No of Mandatory HMOs column refers to the number of properties with 5 or more deposits.

Dataset	Unique Addresses	Precision (%)	Recall (%)
DPS	2,981	99.1	100
MyDeposits	886	76.3	98.5
TDS	1,490	NA	NA

Table 6: Table showing information regarding the total number of clusters found in the datasets for MyDeposits and DPS data (Properties with at least 1 tenant). Each instance in the TDS dataset represents a unique property, therefore the total number of properties represents in the TDS dataset was equal to the number of isntances (1,490).



(a) (b)

Figure 9: Graph (a): P-R plot for MyDeposits Dataset Graph (b): P-R plot for DPS dataset. These graphs were used to manually interpret the optimal trade off between precision and recall for the address clustering tool.

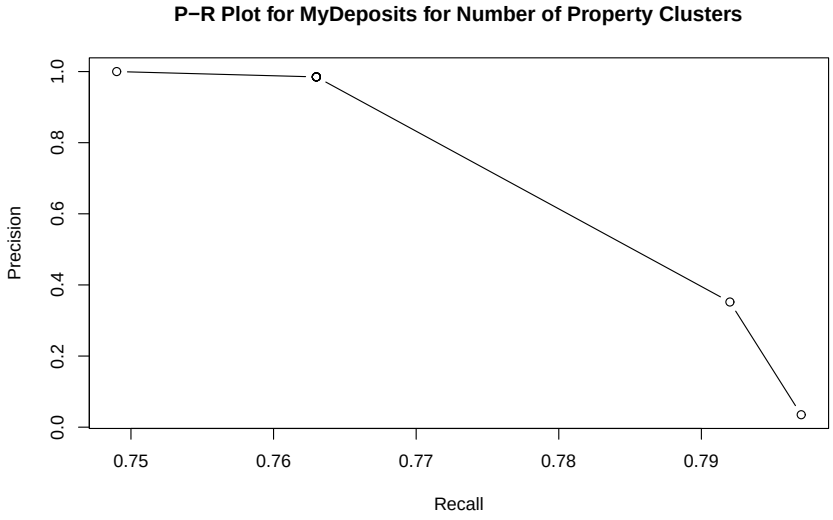


Figure 10: P-R plot for MyDeposits data for the total number of addresses represented in the dataset. This graph was used to find the optimal trade off between precision and recall.

5.2 Advertisement Classification Results

The DT model built is shown in Figure 11. The DT model built from the training data was tested against the the validation data using the 'predict' command. Table 7 shows the confusion matrix. The classification accuracy for the model was 100%.

	Predicted Negative	Predicted Positive
Actual Negative	15	0
Actual Positive	0	20

Table 7: Confusion matrix for advertisement classification model.

The printcp() command was used to gain information on the complexity parameter (cp) and the cross validation error was plotted as a function of cp and tree size (Figure 12) using data from table 8. The cross-validation error is computed using the procedure shown in Appendix C.

N-Split	CP	Relative Error (%)	X-Error (%)	X-std
0	0.8	1.0	1.0	0.2
1	0.0	0.2	0.467	0.16
2	-1.0	0.2	0.467	0.16

Table 8: A table of optimal prunings based on complexity parameter. CP minimizes after the 1st split.

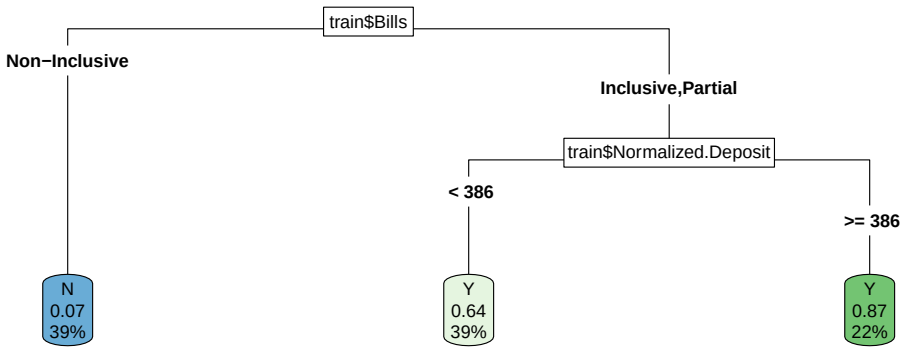


Figure 11: Tree model created in R for detected non-compliant HMO properties.

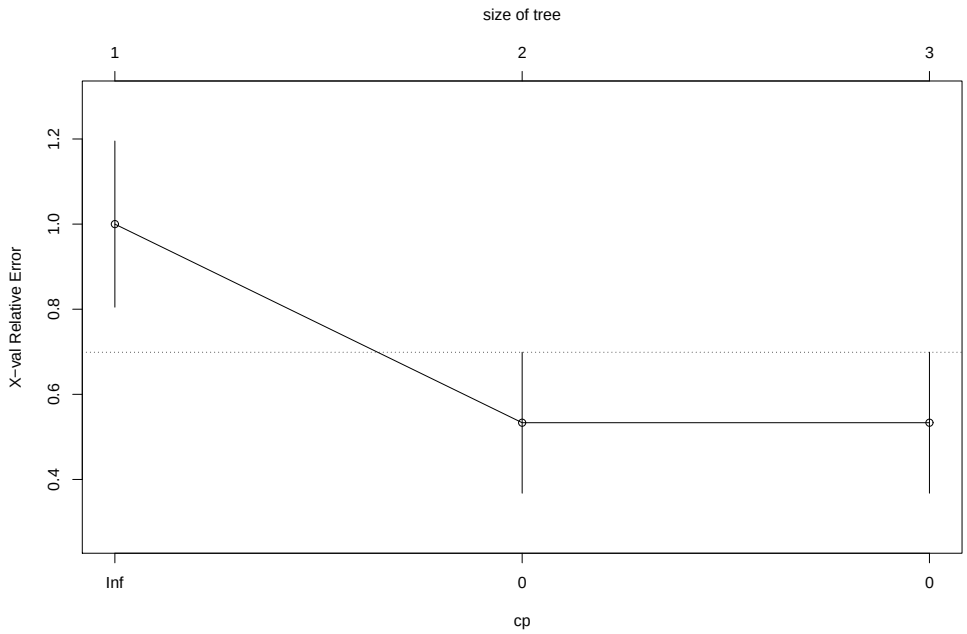


Figure 12: Cross validation error (X-val Relative Error) as a function of complexity parameter and tree size. All variables have no units.

6 Discussion & Evaluation

Table 9 shows a summary of how the CRISP-DM methodology was applied to this project. The main issue identified when modelling HMOs was that they present themselves as a rare event. The literature review was used to inform the methodology design so that this could be accounted for. In the address clustering implementation, it was found that computing string similarities required substantial computational power. The rent deposit records contain 27,272,805 pairs of unique addresses to calculate string similarities on, so the initial data was sampled to make the solution more scalable. The Levenshtein distance was chosen to compute similarities between property addresses because it was superior to other distance functions discussed in the literature review. The Levenshtein distance allows for insertions, deletions and replacements of string characters, so would work better than Hamming distances and longest common subsequences when using data which may contain typing errors such as unnecessary characters.

The second research objective was initially to develop a rare event classification model to discern between HMO and non-HMO property advertisements; however, early prototype models and preliminary testing found that the models were prone to overfitting and were frequently dominated by single variables such as rent deposit cost. Measures were taken to normalize these variables such as conducting dimensional analysis and changing variables such as 'rent deposit' and 'rent per month' into units of £s per bedroom, but this technique did not work. Therefore, the research objective was retrospectively changed to 'Conduct research to test whether it is feasible to develop a classification model for predicting the likelihood of a property advertisement being a HMO. The tree model built performed with 100% accuracy, but this is indicative of overfitting. The embedded feature selection of rpart fitted to the type of bills the property had and discarded the other features collected. This is possibly due to having a sample which is not representative of property advertisements in the Bedfordshire area.

Phase	Implementation
Business Understanding	The motivations section during the introduction of the report contributed to the formulation of research objectives 1 and 2. Furthermore, the background section discussed the theoretical underpinnings of the project and knowledge gained there informed the methodology for achieving the research objectives. Consultations with the Privately Rented Sector Housing Manager at CBC, Jonathan Arnold were also used to contribute to business understanding.
Data Understanding	Manual inspection of rent deposit records was conducted to assess data quality which was documented in sections 3.1.1 and 3.1.2.
Data Preparation	Through inspection of the rent deposit records, it was concluded that the records lacked a consistent schema and a measure was taken to normalize the data. The data collection phase of the property advertisement classification experiment undersampled the majority class.
Modelling	The rent deposit records were modelled using the dedupe Python library. The 'rpart' library was used to model the property advertisments.
Evaluation	The clustering tool was evaluated using precision and recall metrics, emphasis was placed on precision. The property advertisement classification model was evaluated using accuracy.
Deployment	The deployment phase involved automating the clustering tool but was not reported on here. No deployment measures were required for the property advertisement research.

Table 9: A table outlining how each phase of the CRISP-DM methodology was implemented in this project.

According to CBC, there are currently 11,000 privately rented properties in the Bedfordshire area. In the introduction of this report, it was reported that the English Housing survey of counties performed in 2008 estimated that there are around 236,000 HMOs in England, approximately 1.1% of privately rented properties; whereas analysis of the 2011 census data report by Cauvain estimates that HMO properties make up approximately 4.3% of privately rented properties. If we model Bedfordshire as an average county, we would expect the number of HMOs to lie within the range of 1.1-4.3% of the total number of privately rented properties. The rent deposit datasets combined contained 7,386 records, representing 5,357 properties and 48.7% of the total number of privately rented properties in Bedfordshire. If we scale down the initial expected proportion of HMO properties (1.1-4.3%) on the order of 0.487, we find a new range which lies between 0.54-2.1%. The ratio of properties to total properties $\frac{64}{11{,}000} \times 100 = 0.581\%$, which is consistent with the expected proportion of HMO properties, based on the English Housing survey and census data. . It was noted in the introduction of this report that some scholars have criticized these analyses for under representing HMO properties. Therefore, a limitation of this evaluation technique is the assumption that data from the UK census and English housing survey is accurate with respect to their reporting on the frequency of HMO properties in the privately rented sector.

6.1 Originality of Work

The work done in this project largely consists of applying knowledge from other contexts to a new context; however, some aspects of the project have dealt with the creation of original knowledge. Previous research on HMOs comes in the form of surveys, questionnaires and anecdotal studies. No published works have discussed detecting HMOs in datasets held by local authorities. HMOs are usually detected through inspections of properties due to complaints sent the council regarding non-related matters. Dedupe was originally designed for document deduplication and entity resolution purposes but was successfully applied to detecting HMO properties from rent deposit datasets. Original data on property advertisements was collected and no attempt has previously been made to detect non-compliant HMO properties on property advertisement websites.

7 Conclusions

The 1^{st} research objective was to develop a method of clustering addresses from rent deposit data to provide a decision support tool for HMO detection which Housing Officers at CBC can review. The results of the address clustering procedure in dedupe provided CBC Housing Officers with actionable results. Letters were sent out to the addresses found to have more that 5 rent deposits records using the letter template shown in appendix D. The outcome of these letters has not been reported here because the recipients were given 1 month to respond, a longitudinal analysis would've been required in order to fully evaluate how many addresses detected by the address clustering tool were HMOs and this would've exceeded the timeframe of the project. In spite of this, it can be concluded that the 1^{st} was achieved because the address clustering procedure performed entity resolution on addresses with high precision.

The 2^{nd} research objective was to test the feasibility of developing a rare-event classification model for predicting the likelihood of a property advertisement being a HMO, which included: selecting candidate features which are easily accessible from property advertisement websites and creating an original dataset containing instances of HMO and non-HMO target variables, along with predictors for the creation of training and validation sets. A CART model was built which was based on a dataset containing a small number of instances which gives it lower population validity, hindering the generalizability of the model. CART models built from small sample sizes are less accurate than CART models built from samples with a greater number of instances; however, this is a common problem amongst most rare-event models where considerations for absolute and relative rarity must be made. Notwithstanding, the model was able to effectively discern between HMO and non-HMO properties with high accuracy, but this was likely due to overfitting, since the model had an accuracy of 100%. The embedded feature selection of the CART model reduced the candidate features to a single variable, overshadowing the other variables which were easily accessible from property advertisements. Therefore, it is likely not possible at this moment, given the limited data, that a rare-event classification model could be developed to detect

non-compliant HMO property advertisements.

7.1 Social, Ethical and Professional Considerations

Data analysis involved in this project was consistent with the General Data Protection Regulation [69], however; false positive HMOs classed by the address clustering tool could raise some social and ethical issues in the form of harassment of Landlords and letting agents who are not operating HMO properties and could also lead to a waste of council time and resources. Measures were taken to negate these issues by placing a high emphasis on precision, which was achieved for both rent deposit datasets. Furthermore, the tool is to be used by professionally trained Housing Officers at CBC, who will be able to further investigate addresses classed as HMOs by the tool using council held data before having to formally make an inquest to the Landlords and letting agents. Classification of property advertisements could potentially pose a similar issue; however, in the context of this report, the model was used as standalone research and will not be actioned on. The address clustering tool was built using python, this raises some professional issues because Housing Officers at CBC are not trained to use python. Measures were taken to automate the tool, so that knowledge of python would not be required for its use, but its not possible to provide a full assessment of long-term usability without exceeding the timeframe of the project.

7.2 Avenues for Future Research

In the introduction, it was established that HMO properties tend to have a greater risks of energy poverty, fires, damp, mould, injuries due to falls between levels and an increased incidence of anti-social behaviour complaints. The council holds additional datasets which could provide further insight into those features in the form of energy performance certifications and anti-social behavioural complaints. The decision support tool could be expanded to indicate whether a detected property also appears in those databases, to highlight properties of greater risk to health and safety and anti-social behaviour. Further to this, the council has suggested that properties associated with food business such as fast food take-aways

with poor food hygiene ratings could have greater risks of fires, so addresses which appear on food business score databases held by the council could be flagged as having a higher risk of fire.

Previous research contained in literature has indicated that classifying HMO properties based on deposit data is limited. For example, a 2016 case study by Green et al carried out semi-structured interviews with Landlords and tenants of HMOs with the objective of documenting the tenant's experiences of living in HMOs and the Landlords experiences with managing a HMOs. The interviewers engaged in discussion with Landlords about their property ownership history, how they select tenants and their personal relationship with them. One Landlord stated: "I do not take a deposit in order to avoid the associated bureaucracy when the tenancy comes to an end" [70]. Notwithstanding their reasons for being dismissive of rent deposits, the result is that such HMOs would not appear on rent deposit databases held by councils; if this method was the sole way of predicting whether a property was a HMO, a number of HMOs would become undetectable. Therefore, a more holistic approach requires alternative methods which do not rely solely on rent deposits for the detection of HMO properties. Counting the number of people eligible to pay council tax at a given property is another technique which could be used.

If future research was to be done on the development of a rare-event classification model for detecting non-compliant HMO properties, one drawback to consider with the research presented here is the limited amount of statistical data available to inform the creation of the dataset used, which made it difficult to evaluate whether the sample was representative, which is likely the cause of overfitting in the model built. Manual measures were taken for data collection, which was inefficient and time consuming. In future, this could be automated via a web crawler. Further to this, the web crawler could be used to gather large amounts of data on property advertisements and in turn, several statistical features could be calculated and used to inform the creation of a more representative sample of property advertisements.

References

[1] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth *et al.*, "CRISP-DM 1.0: Step-by-step data mining guide," *SPSS inc*, vol. 16, 2000.

[2] J. Han, J. Pei, and M. Kamber, "Data mining: concepts and techniques." Waltham, USA: Elsevier, 2012, ch. 9.

[3] "Drawing Neural Networks with Tikz," https://tex.stackexchange.com/questions/153957/drawing-neural-network-with-tikz, Accessed.

[4] J. Han, J. Pei, and M. Kamber, "Data mining: concepts and techniques." Waltham, USA: Elsevier, 2012, ch. 8, p. 366.

[5] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection." in *MAICS*, 2011, pp. 120–127.

[6] Housing Act (1985). Accessed 14/07/2019. [Online]. Available: https://www.legislation.gov.uk/ukpga/1985/68/contents

[7] Housing Act (2004). Accessed 14/07/2019. [Online]. Available: http://www.legislation.gov.uk/ukpga/2004/34/contents

[8] C. Barratt, C. Kitcher, and J. Stewart, "Beyond safety to wellbeing: How local authorities can mitigate the mental health risks of living in houses in multiple occupation," *Journal of Environmental Health Research*, vol. 12, no. 1, pp. 39–51, 2012.

[9] J. Cauvain (nee Viitanen) and D. Weatherall, "Housing in multiple occupancy: Energy issues and policy," 06 2014.

[10] A. Layard, "Law and localism: the case of multiple occupancy housing," *Legal Studies*, vol. 32, no. 4, p. 551–576, 2012.

[11] Office of the Deputy Prime Minister, "Housing, Planning, Local Government and the Regions Committee," 2008.

[12] T. Gibley, "Housing Enforcement Policy v2.4," *Central Bedfordshire Council, Social Care, Health and Housing*, 2019.

[13] J. Cauvain and S. Bouzarovski, "Energy vulnerability in multiple occupancy housing: a problem that policy forgot," *People, Place & Policy Online*, vol. 10, no. 1, pp. 88–106, 2016.

[14] Housing and Planning Act 2016. Accessed 05/08/2019. [Online]. Available: http://www.legislation.gov.uk/ukpga/2016/22/contents/enacted

[15] J. Arnold and S. Breheny, "Housing Enforcement: Financial Penalty Policy," *Central Bedfordshire Council, Social Care, Health and Housing*, 2019.

[16] "Evaluation of the impact of hmo licensing and selective licensing," in *Department of Communities and Local Government (CLG)*. DCLG London, 2010.

[17] Y. Li, L. Maguire, M. McCann, A. Johnston *et al.*, "Prediction performance improvement for highly imbalanced monitoring data," in *Condition Monitoring and Machinery Failure Prevention Technologies, 7th International Conference 2010 (2 Vols)*. British Institute of Non-Destructive Testing (BINDT), 2010.

[18] M. Rogalewicz and R. Sika, "Methodologies of knowledge discovery from data and data mining methods in mechanical engineering," *Management and Production Engineering Review*, vol. 7, no. 4, pp. 97–108, 2016.

[19] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. Citeseer, 2000, pp. 29–39.

[20] A. Azevedo, I. R. Lourenço, and F. Santos, Manuel, "KDD, SEMMA and CRISP-DM: a parallel overview," *IADS-DM*, 2008.

[21] D. Steinberg and P. Colla, "Cart: classification and regression trees," *The top ten algorithms in data mining*, vol. 9, p. 179, 2009.

[22] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 7–19, 2004.

[23] G. M. Weiss and H. Hirsh, "Learning to predict rare events in event sequences." in *KDD*, vol. 98, 1998, pp. 359–363.

[24] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[25] C. Ferri, J. Hernández-Orallo, and P. A. Flach, "A coherent interpretation of auc as a measure of aggregated classification performance," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 657–664.

[26] M. Herland, R. A. Bauder, and T. M. Khoshgoftaar, "The effects of class rarity on the evaluation of supervised healthcare fraud detection models," *Journal of Big Data*, vol. 6, no. 1, p. 21, 2019.

[27] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, 2000, pp. 111–117.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[29] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[30] D. Pelleg and A. W. Moore, "Active learning for anomaly and rare-category detection," in *Advances in neural information processing systems*, 2005, pp. 1073–1080.

[31] C. Charras and T. Lecroq, *Handbook of Exact String Matching Algorithms*, 01 2004.

[32] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[33] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 8, pp. 707–710, 1966.

[34] C. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval." Cambridge University Press, 2009, ch. 3, pp. 58–60.

[35] D. Hirschberg, "Serial computations of levenshtein distances," pp. 123–141, 1997.

[36] A. Bookstein, V. Kulyukin, and T. Raita, "Generalized hamming distance," *Information Retrieval*, vol. 5, 10 2002.

[37] B. Hussain, O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller, "An evaluation of clustering algorithms in duplicate detection," *University of Toronto Technical Report*, 2013.

[38] C. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval." Cambridge University Press, 2009, ch. 17, pp. 377–401.

[39] J. Han, J. Pei, and M. Kamber, "Data mining: concepts and techniques." Waltham, USA: Elsevier, 2012, ch. 10, pp. 444–490.

[40] P.-N. Tan, M. Steinbach, and V. Kumar, "Introduction to data mining." Pearson, 2006, ch. 8.

[41] J. Han, J. Pei, and M. Kamber, "Data mining: concepts and techniques." Waltham, USA: Elsevier, 2012, ch. 8, p. 327.

[42] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007, pp. 3–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1566770.1566773

[43] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

[44] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of clinical epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[45] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.

[46] E. Cakır and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[47] W. Choe, O. Ersoy, and M. Bina, "Neural network schemes for detecting rare events in human genomic dna," *Bioinformatics*, vol. 16, pp. 1062–1072, 12 2000.

[48] G. A. Carpenter and S. Grossberg, "A self-organizing neural network for supervised learning, recognition, and prediction," *IEEE Communications Magazine*, vol. 30, no. 9, pp. 38–49, Sep. 1992.

[49] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017, pp. 80–84.

[50] T. M. Mitchell *et al.*, "Machine learning," *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 52–59, 1997.

[51] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[52] J. Hebert, "Predicting rare failure events using classification trees on large scale manufacturing data with complex interactions," in *2016 IEEE International Conference on Big Data (Big Data)*, Dec 2016, pp. 2024–2028.

[53] J. Hintze, "NCSS Statistical Software: Logistic Regression," *NCSS, Kaysville, UT*, 1998.

[54] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[55] R. Williams, "Analyzing Rare Events with Logistic Regression," *University of Notre Dame*, 2018.

[56] R. Puhr, G. Heinze, M. Nold, L. Lusa, and A. Geroldinger, "Firth's logistic regression with rare events: accurate effect estimates and predictions?" *Statistics in Medicine*, vol. 36, no. 14, pp. 2302–2317, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.7273

[57] K. M. Leung, "Naive Bayesian Classifier," *Finance and Risk Engineering*, 2007.

[58] L. C. van der Gaag and A. Capotorti, "Naive bayesian classifiers with extreme probability features," in *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, ser. Proceedings of Machine Learning Research, V. Kratochvíl and M. Studený, Eds., vol. 72. Prague, Czech Republic: PMLR, 11–14 Sep 2018, pp. 499–510. [Online]. Available: http://proceedings.mlr.press/v72/van-der-gaag18b.html

[59] K. Kira, L. A. Rendell *et al.*, "The feature selection problem: Traditional methods and a new algorithm," in *Aaai*, vol. 2, 1992, pp. 129–134.

[60] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[61] J. Biesiada and W. Duch, "Feature selection for high-dimensional data—a pearson redundancy based filter," in *Computer recognition systems 2*. Springer, 2007, pp. 242–249.

[62] I. Guyon, "Practical feature selection: from correlation to causality," *Mining massive data sets for security: advances in data mining, search, social networks and text mining, and their applications to security*, pp. 27–43, 2008.

[63] J. Han, J. Pei, and M. Kamber, "Data mining: concepts and techniques." Waltham, USA: Elsevier, 2012, ch. 3, p. 95.

[64] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, ser. ICML '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 412–420. [Online]. Available: http://dl.acm.org/citation.cfm?id=645526.657137

[65] C. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval." Cambridge University Press, 2009, ch. 5, pp. 97–100.

[66] F. Gregg and E. Derek. (2015) Dedupe. Accessed 06/08/2019. [Online]. Available: https://github.com/dedupeio/dedupe

[67] D. Eder and G. Forest, "dedupe documentation, Release 1.9.4," 2019, Accessed: 08/08/2019. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/dedupe/latest/dedupe.pdf

[68] T. M. Therneau, E. J. Atkinson *et al.*, "An introduction to recursive partitioning using the rpart routines."

[69] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. Accessed 08/09/2019. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[70] G. Green, C. Barratt, and M. Wiltshire, "Control and care: landlords and the governance of vulnerable tenants in houses in multiple occupation," *Housing Studies*, vol. 31, no. 3, pp. 269–286, 2016.

A Python Script for Address Clustering

The following is the general Python script used to cluster property addresses, some aspects such as filenames and fields are changed according to the rent deposit dataset being used. In this case, property addresses from MyDeposits data are being clustered. Lines 54 to 62 and 94 to 131 were copied from the dedupe documentation.

```
1  import pandas as pd
2  from future.builtins import next
3  import os
4  import csv
5  import re
6  import dedupe
7
8  # Import Rent Deposit Data
9
10 df = pd.read_csv("D:\MSc Project\Data\Rent_Deposits\MyDeposits.csv",
       header=6,index_col=False,usecols
       =[1,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18])
11 df1 = pd.DataFrame(df)
12
13 # Find postcodes which appear more than 3 times
14
15 sub_df = df1[['Rental_Property__Postcode','Rental_Property_Line1']]
16
17 df_sum = sub_df.groupby('Rental_Property__Postcode')\
18         ['Rental_Property_Line1'].count()\
19         .reset_index(name="count")
20
21 df_sum1 = df_sum[df_sum['count'] >= 3].reset_index()
22 print(df_sum1['count'].sum())
23
```

```
df_sum2 =  df_sum1['Rental Property  Postcode']
reddep = df1[~df1['Rental Property  Postcode'].isin(df_sum2) == False].
    reset_index()

print(reddep)
reddep.to_csv(r'D:\MSc Project\Data\Rent_Deposits\reddep3.csv')

input_file = "D:\\MSc Project\\Data\\Rent_Deposits\\reddep3.csv"
output_file = "D:\\MSc Project\\Data\\Rent_Deposits\\reddep4.csv"
settings_file = 'csv_example_learned_settings1'
training_file = "D:\\MSc Project\\Data\\Rent_Deposits\\trialtraining3.
    json"

# ## Setup

def NormalizeAddress(addressline):
#Normalize rental property addresses so that only text and numbers
    remain #with no empty spaces when deletions occur
        addressline = re.sub(',',' ', addressline)
        addressline = re.sub('/',' ', addressline)
        addressline = re.sub('[^A-Za-z0-9]+', ' ', addressline)
        addressline = re.sub('\[',' ', addressline)
        addressline = re.sub('\]',' ', addressline)
        addressline = re.sub('\(|\)' ' ', addressline)
        addressline = re.sub('\[[^]]*\]', ' ', addressline)
        addressline = re.sub(r'[?|$|.|!]',r' ',addressline)
        addressline = re.sub('  +', ' ', addressline)
        addressline = re.sub('\n', ' ', addressline)
        addressline = addressline.strip().strip('"').strip("'")\
        |       .lower().strip()
        if not addressline:
        |       addressline = None
```

```
53          return addressline
54
55  def readData(filename):
56          data_d = {}
57          with open(filename) as f:
58          |       reader = csv.DictReader(f)
59          |       for row in reader:
60          |       |       clean_row = [(k, NormalizeAddress(v)) for (k, v
                ) in row.items()]
61          |       |       row_id = int(row['newindex'])
62          |       |       data_d[row_id] = dict(clean_row)
63  return data_d
64
65  data_d = readData(input_file)
66
67  fields = [
68          {'field': 'Rental Property Line1', 'type': 'String'},
69          {'field': 'Rental Property  Postcode', 'type': 'String'},
70          ] #defines fields to cluster on, different for each dataset
71
72  deduper = dedupe.Dedupe(fields)
73
74  deduper.sample(data_d, 1500)
75
76  dedupe.consoleLabel(deduper)
77  deduper.train(recall=0.5)
78
79  with open(training_file, 'w') as tf:
80  deduper.writeTraining(tf)
81
82  with open(settings_file, 'wb') as sf:
83  deduper.writeSettings(sf)
```

```
84
85 threshold = deduper.threshold(data_d, recall_weight=0.5)
86
87 print('clustering...')
88 clustered_dupes = deduper.match(data_d, threshold)
89 print('# duplicate sets', len(clustered_dupes))
90
91 #The following section (94 to 131) is taken from dedupes documentation
      and is not my own work
92 # https://github.com/dedupeio/dedupe
93 # It writes the original data back out to a CSV with a new column
      called
94 # 'Cluster ID' which indicates which records refer to each other.
95 cluster_membership = {}
96 cluster_id = 0
97 for (cluster_id, cluster) in enumerate(clustered_dupes):
98 |       id_set, scores = cluster
99 |       cluster_d = [data_d[c] for c in id_set]
100 |       canonical_rep = dedupe.canonicalize(cluster_d)
101 |       for record_id, score in zip(id_set, scores):
102 |       cluster_membership[record_id] = {
103 |       |       "cluster_id": cluster_id,
104 |       |       "canonical_representation": canonical_rep,
105 |       |       "confidence": score
106 |       }
107 singleton_id = cluster_id + 1
108
109 with open(output_file, 'w') as f_output, open(input_file) as f_input:
110         writer = csv.writer(f_output)
111         reader = csv.reader(f_input)
112         heading_row = next(reader)
113         heading_row.insert(0, 'confidence_score')
```

```
        heading_row.insert(0, 'Cluster ID')
        canonical_keys = canonical_rep.keys()
        for key in canonical_keys:
        |       heading_row.append('canonical_' + key)
        writer.writerow(heading_row)
        for row in reader:
        |       row_id = int(row[0])
        |       if row_id in cluster:
        |       |       cluster_id = cluster[row_id]["cluster id"]
        |       |       canonical_rep = cluster[row_id]["canonical"]
        |       |       row.insert(0, cluster[row_id]['confidence'])
        |       |       row.insert(0, cluster_id)
        |       |       for key in canonical_keys:
        |       |       |       row.insert(0, None)
        |       |       |       row.insert(0, singleton_id)
        |       |       |       singleton_id += 1
        |       |       for key in canonical_keys:
        |       |       |       row.append(None)
        |       writer.writerow(row)

# Properties with 5 or more records
reddep4 = pd.read_csv("D:\\MSc Project\\Data\\Rent_Deposits\\reddep4.
    csv")
reddep_sum = reddep4.groupby('Cluster ID')\
['Rental Property Line1'].count()\
.reset_index(name="count")
reddep_sum1 = reddep_sum[reddep_sum['count'] >= 5].reset_index()
print(reddep_sum1)
reddep_sum2 = reddep_sum1['Cluster ID']
reddepfinal = reddep4[~reddep4['Cluster ID'].isin(reddep_sum2) == False
    ].reset_index()
reddepfinal.to_csv(r'D:\MSc Project\Data\Rent_Deposits\reddep5.csv')
```

```
144
145 # Properties with 3 or 4 records
146 reddep6 = pd.read_csv("D:\\MSc_Project\\Data\\Rent_Deposits\\reddep4.
        csv")
147 reddep_sum = reddep4.groupby('Cluster_ID')\
148 ['Rental_Property_Line1'].count()\
149 .reset_index(name="count")
150 reddep_sum3 = reddep_sum[reddep_sum['count'] >= 3 && <5].reset_index()
151 print(reddep_sum3)
152 reddep_sum4 = reddep_sum3['Cluster_ID']
153 reddepfinal2 = reddep6[~reddep6['Cluster_ID'].isin(reddep_sum4) ==
        False].reset_index()
154 reddepfinal.to_csv(r'D:\MSc_Project\Data\Rent_Deposits\reddep7.csv')
```

B R Script for Property Advertisement Classification

```
setwd('D:\\MSc Project\\Data\\Advertisements')

prelim3 <- read.csv('prelim3.csv', header=T)

str(tree_data); class(tree_data); mode(tree_data)

pairs(prelim3, col=prelim3$..HMO)



library('rpart')
library('rpart.plot')

set.seed(1234) #seed for training set
pd <- sample(2, nrow(prelim3),replace=TRUE, prob=c(0.8,0.2))
pd
train <- prelim3[pd==1,]
validate <- prelim3[pd==2,]
tree_fit <- rpart(train$..HMO ~ train$Normalized.Deposit
+ train$EPC.Rating+ train$Bills + train$Furnishing,
data = train, method = 'class', cp='-1')

plotcp(tree_fit);printcp(tree_fit)

rpart.plot(tree_fit)



pred <- predict(tree_fit, train, type="class")
tab <- table(pred, train$..HMO)
```

```
print(tab)
sum(diag(tab))/(sum(tab))

test_predict <- table(predict(tree_fit, newdata=validate, type=
   "class"), validate$..HMO)
sum(diag(test_predict))/(sum(test_predict)) # classification
   accuracy
1-sum(diag(tab))/sum(tab) # classification error
```

C Cross-Validation Procedure

Cross-validation procedure taken from "An Introduction to Recursive Partitioning Using the RPART Routines".

1. Fit the full model on the data set
   compute $I_1, I_2, ..., I_m$
   set $\beta_1 = 0$
   $\beta_2 = \sqrt{\alpha_1 \alpha_2}$
   $\beta_3 = \sqrt{\alpha_2 \alpha_3}$
   $\vdots$
   $\beta_{m-1} = \sqrt{\alpha_{m-2} \alpha_{m-1}}$
   $\beta_m = \infty$
   each β_i is a 'typical value' for its I_i

2. Divide the data set into s groups $G_1, G_2, ..., G_s$ each of size s/n, and for each group separately:

   - fit a full model on the data set 'everyone except G_i' and determine $T_{\beta_1}, T_{\beta_2}, ..., T_{\beta_m}$ for this reduced data set,
   - compute the predicted class for each observation in G_i, under each of the models T_{β_j} for $1 \leq j \leq m$,
   - from this compute the risk for each subject.

3. Sum over the G_i to get an estimate of risk for each β_j. For that β (complexity parameter) with smallest risk compute T_β for the full data set, this is chosen as the best trimmed tree.

D Declaration of HMO Notice Attachment

Declaration of a House in Multiple Occupation
Housing Act 2004, Section 255

To:
At:

The property known as [insert address] is declared a House in Multiple Occupation.

On [insert date] [insert name of local authority], the local housing authority decided that this [building] [part of the building] at [insert address] is a House in Multiple Occupation and that this notice of declaration be served.

If no appeal is made against this notice then this declaration shall become operative on [insert date], being not less than 28 days after the decision to serve this notice was made.

You have the right of appeal against this decision to the Property Chamber of the First Tier Tribunal (PC) Service.

The office of the First Tier Tribunal (Property Chamber) for the region in which [insert name of local authority] is located and to which appeals should be made is [insert address, telephone number and email address of appropriate office] and more information can be provided by that office.

More information can be found at http://www.justice.gov.uk/tribunals/residential-property where it is also possible to download the appropriate forms.

You must contact the Property Chamber of the First Tier Tribunal (PC) Service within 28 days of the decision being made. In this instance that is by the [insert date + 28 days].

Dated this day of 20

Signed:

Designation:

All correspondence and enquiries should be made to [insert]