2023-12-18

Early Identification of Youth at Risk of Long Term Emergency Homeless Shelter Use: An Evaluation of Interpretable Machine Learning models

Annaa, Osman Jakpa

UNIVERSITY OF CALGARY

Early Identification of Youth at Risk of Long Term Emergency Homeless Shelter Use: An Evaluation of Interpretable Machine Learning models.

by

Osman Jakpa Annaa

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING

CALGARY, ALBERTA
DECEMBER, 2023

Abstract

Homelessness is a serious violation of one's dignity, and youth who are long-term shelter users are particularly vulnerable members of a vulnerable demographic. The commitment to prevent and eliminate homelessness, particularly among the youth, is a shared responsibility. Programmes aiming at providing homeless people with permanent housing, mostly identify people who have lived with the condition for an extended period of time for support. Allowing young people to be homeless for an extended period of time before intervening, exposes them to several kinds of hardships on the streets. Early identification of youth at risk of becoming a long term shelter user is a proactive and a more humane way of addressing the problem. Machine learning is brought forth as a tool to augment the expertise of shelter staff in identifying youth at risk of long-term shelter use. Machine learning algorithms are utilised to predict youth at risk of long-term shelter use with the clients' first 30, 60, 90, 120, or 180 days of shelter access records. A real time program delivery approach was incorporated in the experiments as a supplement to existing other methods in fighting homelessness. Interpretable machine learning models capable of ultimately producing classification rules in DNF format are evaluated. The level of control over the complexity of the generated rules, coupled with statistical evaluation metrics are employed in the evaluation.

Preface

This thesis is an original work by the author. No part of this thesis has been previously published.

Acknowledgements

My most profound appreciation goes to Professor Geoffrey Messier, my esteemed supervisor, for his time, effort, patience and understanding in helping me succeed in my studies. His vast wisdom, wealth of experience and extensive knowledge in the subject matter have inspired me throughout my research. I would like to thank the Department of Electrical and Software Engineering for providing me with the resources to pursue my graduate studies. Friends, lab mates, colleagues, and research team are all appreciated for the fun times we had working and socializing together. I would also like to thank everyone who has been there for me emotionally and intellectually throughout this period. I would like to thank my parents who set me off on the road to this MSc a long time ago. For my kids, sorry for not being there for you whilst I wrote this thesis! And for my lovely wife Habibata, thanks for your unwavering support over the past few years. To conclude my sincere gratitude goes to Allah, for providing me with life and strength.

Table of Contents

List of Tables

List of Figures

List of Symbols, Abbreviations, and Nomenclature

Abbreviations	Definition
I.T	Information Technology
AI	Artificial Intelligence
ML	Machine Learning
GPU	Graphics Processing Unit
XAI	Explainable Artificial Intelligence
DARPA	Defense Advanced Research Projects Agency
LIME	Local Interpretable Model-agnostic
FLS	Fuzzy Logic System
LRP	Layer-wise Relevance Propagation
DNF	Disjunctive Normal Form
RNN	Recurrent Neural Network
MLP	Multilayer Peceptron
HIFIS	Homeless Individuals and Families Information System
EDA	Exploratory Data Analysis
DI Centre	Calgary Drop-In Centre
CFREB	Conjoint Faculties Research Ethics Board
ID	Identity
EMS	Emergency Medical Service
CPS	Calgary Police Service
GoC	Government of Canada
GoA	Government of Alberta

LT	Long-term clients
CART	Classification and Regression Trees
GOSDT	Generalised and Scalable Optimal Sparse Decision Trees
EXPLORE	Exhaustive Procedure for LOgic-Rule Extraction
NP-Complete	Nondeterministic Polynomial-Complete
TDIDT	Top-Down Induction of Decision Tree
AUC	Area Under the ROC Curve
ROC	Receiver Operating Characteristic Curve
OSDT	Optimal Sparse Decision Trees
CORELS	Certifiably Optimal RulE ListS
TP	True Positives
FP	False Positives
TN	True negatives
FN	False Negatives

Symbol	**Definition**
f	feature
o	Operator
v	Value
x	Example
y	Label of example
n	nth example
l	Literal
t	Term
r	Rule
R	Rule Complexity
P_j	Probability of samples belonging to class j
c	Number of classes
F	Feature names
P	Performance measure
n	EXPLORE rule length

C	Performance Constraints
FO	Feature Operator Pair
V	Value list
T	Term tuple

Epigraph

It's hard being homeless at any age, but at 16 years old? I can't even imagine. When you're a homeless teen, how do you build a future or have any sort of life?

— Stacey Dooley

The biggest misconception about the homeless is that they got themselves in the mess — let them get themselves out. Many people think they are simply lazy. I urge those to make a friend at a local mission and find out how wrong these assumptions are.

— Ron Hall

Helping others, serving others — this is the real meaning of life.

— Dalai Lama

Chapter 1

Introduction

This research work was aimed at identifying the best algorithm for inducing effective rules that predict clients at risk of long term homelessness from an administrative homeless shelter data while giving full control to the level of rule complexity. In addition, the optimal way to apply these algorithms in a shelter setting was also investigated. This Chapter presents the background to the study and places the significance of the research in context. The rest of the Chapter is organised as follows: In Section 1.1, the need for this work is presented. Section 1.2 provides the motivation for the thesis. In Section 1.3, the importance of leveraging machine learning in the nonprofit space is presented. Section 1.4 provides reasons why we should consider interpretability when dealing with a vulnerable group in a nonprofit setting. In Section 1.5, the importance of giving control to the complexity of rules generated is presented. In Section 1.6, a review of previous work on the problem being addressed is provided. Section 1.7 provides the main contributions of this thesis. Finally, Section 1.8 provides the outline for the rest of the work.

1.1 Homelessness

Homelessness persist as a serious societal problem, affecting millions of people in high income countries like Canada despite the economic boom in recent decades. In [2], it is recorded that 35,000 Canadians experience homelessness on any given night and a minimum of 235,000 Canadians experience homelessness each year. Under article 25 of the 1948 Universal Declaration of Human Rights and in article 11.1 of the 1966 International Covenant on Economic, Social and Cultural Rights; housing is a right to an adequate standard of living [3]. Housing provides stability, security and a place for our social and emotional lives. The emergency shelter system plays a vital role in providing an adhoc place for the homeless to access services like sleep, food and health. It is one of the primary point of contacts to persons that lose their homes. There is an increasing strain on Canada's emergency shelters. On average Canada's emergency shelters operated at 82.7% full capacity in 2005 and 92.4% full capacity by 2014 [2]. At an emergency shelter, there were reportedly 118,759 homeless individuals and families in 2019. Roughly 14,400 people reside in shelters on a typical night [4].

Research has shown that an enormous number of youth in Canada are connected to the streets. About 20% of the homeless population in Canada are youth, an average of 35,000 to 40,000 of youth experience homelessness in a year and 6,000 to 7,000 on a given night [2] [5]. Being homeless at a young age exposes them to several adversities: declining mental health, unemployment, criminal victimization, etc. Homeless youth have a high prevalence of mental problems, with depression, PTSD, and suicidal behaviours being widespread [6] [7]. Youth living with homelessness have mortality rates that are 12-40 times higher than the general population [8], with suicide being the major cause of death [9].

People experiencing homelessness are transitioned into stable homes through initiatives such as the housing first program. Individuals are provided with permanent housing first, then accompanied with the needed supports. People need not to be "ready", thus; meet certain preconditions (i.e. sobriety or abstinence) before they could be allocated permanent housing [10]. Persons living with chronic homelessness are mostly given priority. However, it is difficult to identify these people who are chronically homeless in a timely manner, allowing them to remain in that situation for several months. Employment and social development Canada [11], defines chronic homelessness as individuals who are currently experiencing homelessness and who meet at least 1 of the following criteria:

- have a total of at least 6 months (180 days) of homelessness over the past year.
- they have recurrent experiences of homelessness over the past 3 years, with a cumulative duration of at least 18 months (546 days).

This definition permits a person to be deemed chronically homeless after staying homeless for at least six months. Because chronically homeless people are usually prioritised, earlier identification boosts a person's chances of finding secure housing sooner.

1.2 Motivation

Allowing young persons to stay in an extended state of homelessness before intervening exposes them to the great adversities on the streets. According to [12] [13], although they are more likely than their housed counterparts to have diagnosed mental health conditions, experience all types of abuse, use alcohol and illicit drugs, engage in high-risk sexual behaviour, and have more contact with the criminal justice system;

homeless youths are incredibly vulnerable to differences in access to or utilisation of healthcare services. In both relative and absolute terms, the rates of sickness and mortality among the homeless are high in comparison to the general population [1]. Table 1.1 highlights the disparities in health conditions between the homeless and the housed.

	Prevalence range in homeless people	General population prevalence
Tuberculosis	0–8%	0·005–0·032%
Hepatitis C	4–36%	0·5–2·0%
HIV	0–21%	0·1–0·6%
Hepatitis B	17–30%	<1%
Scabies	4–56%	<1%
Body louse	7–22%	<1%
Bartonella quintana	2–30%	<1%

Table 1.1: Prevalence of infectious diseases in homeless people compared to the general population [1]

Homelessness is not only cruel to persons experiencing its affliction but it is also very expensive to the government. Homelessness cost the Canadian economy 7.05 billion dollars annually, up from 4.5 to 6 billion dollars in 2007 [14]. When compared to placing them in social or supportive housing, the expense of housing someone in the shelter system is significantly higher. The annual cost of supportive and transitional housing is between 13,000 and 18,000 dollars, whereas the cost of emergency shelters ranges from 13,000 to 42,000 dollars and affordable housing without supports a mere 5,000 to 8,000 dollars [15]. Housing someone is not only more affordable, but it is also considerably more humane. The longer a person is homeless, the more likely it is that their bodily and mental health will decline and that they will die young.

Long term clients are the minority class that use shelter services the most. While being few in number, these people heavily rely on emergency services in the homeless community as well as in the fields of health, criminal justice, and social services [14]. Since 2005, there has been a 72.2% increase in the percentage of people who stay in shelters for 31 days or more. All shelters had a 92.3% average nightly occupancy in 2019; family shelters had a 104.2% occupancy rate [4].

Early identification of long term clients for supportive housing will reduce the strain on the shelter system and make it more accessible to transitional clients. As the saying goes prevention is better than cure. The focus to addressing long term or chronic homelessness is shifting from provision of temporal solutions to a more permanent solution like the housing first program. These programs are still not sufficient since persons live with homelessness for long periods before been identified and moved to stable housing. It is imperative that these efforts are augmented, by helping identify at an early stage who may be at particular risk of long term homelessness to mitigate or minimise that risk.

1.3 Machine Learning for Nonprofits

It is practically impossible for a human to look through large sums of data finding patterns that can lead to a future event. Machine learning is a branch of artificial intelligence that allows machines to learn useful hidden patterns within historical data and then predict future outcomes with a new data set. It is important to note that machine learning in the nonprofit space is intended to be an aid that compliments the human expertise. They will make suggestions based on past data and the human has the ultimate decision of whether or not to follow those suggestions. It helps operationalize the expertise of staff in a way that would have taken hours before. It analyzes the data and turn it into actionable outputs.

Most machine learning models are very computationally expensive: this is a critical issue for non-profit organisations relying on simple information technology (I.T.) resources. Deep learning applications' computational requirements heavily rely on advancements in computing power. Deep learning requires powerful computers to function. In an over-parameterized situation, computational needs expand as the square of the amount of data points, and the cost of training a deep learning model scales with the product of the number of parameters and the number of data points [16]. The training costs, regardless of whether the deep neural network training is done on locally provided resources or in the cloud is exorbitant. The cost of training a deep neural network on GPUs can quickly approach the tens of thousands of dollars, and modern GPUs provisioned on the cloud for deep learning frequently cost several dollars per hour [17]. Despite the immense benefits of leveraging on machine learning, it is essential to consider cost when choosing a model since the nonprofit organizations lack the compute power to handle most of these algorithms.

When developing machine learning solutions for social issues, it can be difficult for it to be accepted, if it will drastically affect the day-to-day operations of the end users. Restructuring the daily operations of a not for profit organisation may not be possible for reasons such as resource, time, and monetary constraints. Secondly individuals experiencing homelessness who interact with the shelter system, begin their journey with the shelter at different times and have varied patterns of interactions. Hence the indicators that may lead to long term shelter use will show up at different times for different clients. We should therefore not wait to make predictions at a unified time for all clients. If the prediction is made much early, some clients will be missed, on the other hand if the prediction is delayed, clients who could have been identified early for housing support will stay in their condition for a prolonged period. None of these scenarios is desirable. Machine learning models can be implemented in a way that will mimic reality and provide no or minimal adjustment to already existing operations. A real time program delivery at the emergency shelter system needs to be factored in to gain the most out of any machine learning solution provided. It is therefore imperative for a real time machine learning approach to predicting at risk of long-term emergency shelter users be considered.

1.4 Interpretability

There are a variety of machine learning models used to provide solutions for societal problems. These ML models however differ in their approach and no one approach suits all problems. Several factors are considered when selecting an ML model for a specific task, one such factor is interpretability. Before we dive into interpretability lets unravel uninterpretability. Uninterpretable models are usually termed "Black Box" models.

"A black box machine learning model is a formula that is either too complicated for any human to understand, or proprietary, so that one cannot understand its inner workings" [18]. Humans, including those who create them, do not understand how variables are combined to make predictions. Black box models frequently predict the right response for the incorrect reason (the Clever Hans effect), leading to superb training but subpar performance during practise. Since deep learning models have achieved such success in the field of computer vision, it is often believed that the best accurate models for any given data science challenge must be complex and opaque.

The quest for interperatability while harnessing the predictive power of deep learning models gave birth to the advent of explainable models (XAI). Explainable models are created to shed light on what a trained model has learned without altering the underlying model. These are post-hoc models and not part of the primary model. Prediction-level and dataset-level explanations are the two basic groups that encompass the majority of post-hoc explanation techniques. Prediction-level explanation techniques, concentrate on shedding light on specific model predictions, such as how certain features or interactions resulted in a given prediction. Approaches at the dataset level concentrate on the broad associations the model has discovered, such as the visual patterns that are connected to a given projected response. Explainable AI is gaining a lot of attention in the machine learning research domain lately. A number of these works include [19][20][21] and notably, the Defense Advanced Research Projects Agency's (DARPA) explainable artificial intelligence program[22]. Commonly used explanation techniques like, local Interpretable Model-agnostic(LIME) and the fuzzy logic system (FLS) which uses if-then rules and linguistic labels to explain the generated model as well as layer-wise relevance propagation (LRP) are receiving attention.

Instead of explaining a 'black box' model's predictions, inherently interpretable machine learning models with high predictive accuracy should be leveraged in sensitive decision making. The idea of interpretability is broad and includes everything from planning the experiment to visualising the outcomes. In this work, interpretability will be viewed via the prism of outcomes visualisation. Interpretability is the degree to which a human can understand the cause of a decision. In other words interpretability is the degree to which a human can cosistently predict the model's results. In general, logical models (tree models and rule based models), linear models, additive models and disentangled neural networks are viewed as intepretable class of machine learning models.

Interpretability can further be broken down into several components including but not exhaustive; sparsity, self-explainability and modularity. **Sparsity:** Large and complex models are typically difficult to comprehend, given that the human cognitive memory can process 7 ± 2 items at a time [18]. Sparsity is achieved by imposing constraints that limit the model's size. There have been a significant amount of work to achieve sparse interpretable ML models. Some of these work include; [23], GOSDT[24], CORELS[25], and EXPLORE[26]. **Self-explainability:** The goal is for a human (the intended audience) to be able to internally simulate and reason about the model's entire decision-making process. This works well when the number of features are few and the underlying relationship is straightforward. The most common examples that can be easily simulated are logical models (tree models and rule-based models). It is crucial to remember that as these models get bigger, they can no longer be simulated. As the model's complexity grows, it becomes increasingly difficult for a human to internally simulate. **Modularity:** The model is regarded as modular if a significant portion(s) of its prediction-making process can be interpreted independently. Generalized additive models are typical examples that satisfies modularity. Sparsity and self-explainability are very important constrains to impose on a model when the intended interpreters are non-technical.

To gain the trust of emergency shelter staff, interpretability is vital in providing solutions to the vulnerable population. The degree to which they can understand the cause of a machine decision is crucial to them accepting it. Why certain decisions or predictions were made is easier to understand, if machine learning model is sparse and self-explainable. It is not enough to make predictions for shelter staff to follow blindly considering the sensitive nature of the group been assisted. One may argue that why not implement a post-hoc explanable model to explain the decisions made by the primary model. It has been shown in [27], that post-hoc explanations are notoriously unreliable; just because an explanation model is dependent on a variable does not imply that the primary model is dependent on that variable. The study also indicated that there have been cases in criminal justice where post-hoc explanations led to incorrect conclusions.

To implement a machine learning solution on a basic information technology (I.T.) system, interpretability becomes binding. As mentioned earlier deep learning models are computationally expensive and need a resourceful I.T. system to get the best out of them. Secondly an extra model (post-hoc explanable model) needs to accompany a "black box" model to explain the predictions made. This comes with an added burden on an already suffocating system. Rules generated by intepretable models when extracted are easy to implement on a simple spreadsheet, hence intepretability comes at no extra cost to any information technology (I.T.) system.

The aim of this work is to provide the shelter staff with a tool that will aid them in making the final decision on who needs housing support the most. No one understands the homeless population much better than the personnel working with them on daily basis. Predictions made by inherently interpretable models are self explanatory. Shelter staff can look at the rules that made the predictions and gain further insight into the problem of homelessness. In conjunction with their own expertise the staff can then make an informed final decision.

1.5 Rule Complexity Control

Rule complexity is directly related to the model's interperatability. Rules can lose their interpretability when the number of literals in the rule becomes a lot. A detail explanation of the concept of literals and rules is provided in Section 3.2. As indicated in Section 1.4, since humans are only capable of managing 7 ± 2 cognitive entities at once, sparsity is frequently employed as a gauge of interpretability [18]. It is therefore imperative that we have control over the complexity of the rules. Different algorithms provides different techniques to control the complexity of the output rules. A comprehensive description of the complexity control mechanisms provided by the algorithms considered for this work is presented in Chapter 4.

To meet the objectives of this work, interperatable machine learning algorithms are further subjected to the following set of questions:

- Complexity; Is the output of the model complicated? Simplicity is golden. There are algorithms that generates simple models that are comparable to complex models.
- Control: What extent of control does the algorithm offer to decide the level of complexity of the output model? The algorithm must lend, if not full control but a high degree of control of the output model's complexity.
- Implementability: How easy is it to implement the output model? As discussed in section 1.3 it is crucial to consider cost when choosing a model since the non-profit organizations lack the compute power to handle most of these algorithms.

Rules in DNF form are the simplest and easiest to comprehend and implement by non technical persons. A DNF rule is a set of unordered relational rules that collectively represent the knowledge captured by the algorithm. A DNF rule is in the form $\{(f_1 > v_1 \ \wedge \ f_2 \leqslant v_2) \ \vee \ (f_3 = v_3)\}$ where "f" is a feature and "v" is a value threshold. This format of model representation makes it readable and straightforward to comprehend. In Section 3.4 the concept of DNF rules is discussed into details. DNF rules are consistent with the objectives of this work and therefore is the ultimate requirement for interperatable algorithms to meet for this work. All algorithms for this work either have to be able to generate rules in DNF format directly or their output rules should be translatable into DNF format.

1.6 Related Work

There have been several studies that focused on addressing sections of the homeless problem using machine learning. Because eviction is substantially connected with homelessness in this study [28], statistical models (regression and machine learning) were used to predict eviction and find highly correlated determinants of eviction. The study assessed the effectiveness of multiple machine learning models in two areas: identifying features correlated with populations at high risk of eviction using demographic and urban development data, and forecasting potentially evicted populations in the following year. In 2021, [29] proposed a machine learning technique for effectively predicting chronic homeless clients of an emergency shelter utilising time-stamped client access data records in shorter time frames. The viability of using an automated recommendation system to properly match persons to homelessness support facilities when they first become homeless was explored by [30]. Machine learning approaches were specifically employed to recommend the particular service facility that a homeless individual can benefit from. Also, [31] looked at ways to enhance the allocation of critically restricted resources in the context of homelessness service provision. Based on responses to a brief screening tool distributed throughout Veterans Health Administration (VHA), [32] developed and tested predictive models of housing instability and homelessness. A 2018 research [33], aimed at uncovering the factors that influence homeless families' readmission and length of stay. Logistic regression models and an unsupervised clustering approach was employed to uncover predictors of re-entry and long-term length of stay. The problem of ”optimal” resource allocation was formulated in [34], given data on homes in need of homeless services using administrative data from a regional homeless system. Allocating available resources in such a way that the estimated odds of household re-entry are as low as possible was the goal of the optimisation problem. The present system's (Housing and Urban Development (HUD)) success in prioritising youth for housing help and predicting youth homelessness after getting housing aid was explored by [35]. While most of these studies were focused on identifying factors contributing to homlessness and effective resource allocation to the homeless population, only a few considered interpretability [32] [33] [34] [35]. Most of the intepretable models relied on decision trees and logistic regression for interpretability. Short decision trees are relatively simple to comprehend but it should be noted that the number of leaves rises rapidly with depth.

”Black box” models accompanied by post-hoc explainable models have seldomly been investigated. The

HIFIS-RNN-MLP [36] is a combination of a recurrent neural network (RNN) and a multilayer perceptron (MLP). This model was developed to; accurately predict the risk of a person becoming chronically homeless 6 months in advance, as well as identify factors that influence their chronic homelessness and the general driving factors of chronic homelessness in London, Canada. However the drawback of this model is that, it is an opaque model that implemented a post-hoc explanable model; Local Interpretable Model-Agnostic Explanations (LIME), to explain the predictions, but as mentioned in Section 1.3 deep learning models are computationaly expensive and doubling it with a post-hoc model makes it impossible for the I.T systems of emergency shelters to support it. Secondly as discussed in Section 1.4, post-hoc explanable models are known for unreliable explanations of predictions.

There have also been decent amount of research done towards early identification of long term homelessness. In 2021, [37] investigated the most effective ways to employ stay/episode threshold tests to identify clients for housing assistance based on their patterns of shelter access; by developing new threshold-based tests for chronic and episodic shelter usage. Also a study [38], showed that a straightforward threshold approach is comparably effective in predicting chronic homelessness to more complex logistic regression and neural network techniques. While [38][37] have demonstrated the benefits of using a simple threshold test over sophisticated machine learning models for identifying chronic homelessness in a resource constraint nonprofit space, then the power of AI is not leveraged to address societal issues. Additionally, inducing rules for predicting chronic homlessness have been explored by [39]. A rule search framework based on the Opus algorithm [40] for identifying persons at risk of becoming chronic shelter clients and referring them to supportive housing programs was presented. This model generates effective rules, but these rules are not in disjunctive normal form (DNF) and the model does not give direct control of the rule complexity. The concept of DNF is discused in Section 3.4. However, non of these studies mention above, paid closer attention to the youth who are particularly more vulnerable among an already vulnerable group.

Further, [41] presented two screening tools for predicting persistent homelessness: an employment model that predicts whether recently unemployed workers will experience persistent homelessness, and a young adult model that predicts whether youth entering adulthood while receiving public benefits will become persistently homeless. Having more than one episode of homelessness (defined as having no address) within a three-year period qualifies a person as ”persistently homeless” in their work. Though, [41] attempted addressing the problem of youth experiencing persistent homelessness, they only focused on a subset of the youth (those receiving public benefits) neglecting a wider population of the youth.

This work seeks to provide a solution that benefits from the power of AI, to induce rules that are; intuitive to shelter staff, effective at predicting long term shelter users, easy to implement on emergency shelters' simple I.T. systems and hand over full control of its complexity. Secondly, there have not been any literature that explored implementing machine learning solutions that incorporates real time program delivery in the context of homelessness. This work therefore seeks to fill this gap in research by considering real time program delivery at the emergency shelter system in our model development.

1.7 Proposal and Contributions

The main contributions of this work are:

- For the first time, these algorithms are being used to detect long-term emergency shelter users with an emphasis on youth.
- An exploration of the best algorithm for producing effective rules in disjunctive normal form while allowing the best control over their complexity. The target is to identify a model that allows the most control over the complexities of the generated rules in DNF format.
- An investigation into the most effective approach for implementing a machine learning model trained on historical data in a real-time emergency shelter system as part of programme delivery. The purpose is to provide an ML solution that can be incorporated into a shelter operation seamlessly.

1.8 Organization of this Work

The rest of this thesis is organized as follows:

Chapter 2: The Data describes the background, composition, attributes, preprocessing, transformation and labelling of the data, and further discusses the EDA performed.

Chapter 3: Concept Review provides an explanation of important concepts that comes in handy in understanding discussions in the later chapters.

Chapter 4: Algorithms Considered provides a high level description of the machine learning algorithms that were considered in this study.

Chapter 5: Results and Discussion discusses the results of this work; starting from rule complexity and complexity control, metric evaluation and generalisation capacity of the models under review.

Chapter 6: Conclusion sums up the work and provides some suggestions for future work.

Chapter 2

The Data

Data is an essential component and food for machine learning. Without data, machine learning is nothing more than a bare machine with no soul or mind. It's the collection of observations or measurements that can be utilised to train a machine learning model. The quality and quantity of data available for training and testing are important factors in influencing a machine learning model's performance. Data can take several forms, including continuous, categorical, and time-series data. Data is used by supervised machine learning algorithms to discover patterns and correlations between input variables and target outputs, which are subsequently used for prediction or classification tasks.

In this chapter, description of the data is provided, the processing done on the data for use in training the models and also an exploratory analysis on the data is provided. Section 2.1 provides a brief background of the data and the makeup of the data. Section 2.2 describes the attributes within the data. Section 2.3 details the data pre-processing and transformation procedure and also how the data was labeled. Section 2.4 provides interesting insights into the data. Finally, the discussions in this Chapter are summarised in Section 2.5.

2.1 Background and Composition of the Data

The Calgary Drop-In Centre (DI centre) is an emergency shelter that provides essential care e.g., health services, housing support to persons experiencing or at risk of experiencing homelessness and so on. Clients may visit the DI centre to access any of the services provided on daily basis. These interactions of clients with the shelter system is recorded with time stamps. Over a period, these interaction records hold useful hidden patterns, when unearthed could help predict future outcomes. This data is stored and is the origin of the data use for this work. This data is then anonymized by the shelter I.T staff. The anonymization process involves erasing or encrypting identifiers that connect an individual to the stored data. The university of Calgary's Conjoint Faculties Research Ethics Board (CFREB), approved the data handling and anonymization protocol for this work.

The data used for this work is a single table with 5,576,433 rows and 43 columns that contains all client interactions with the Calgary Drop-In Centre that have been anonymised. The raw dataset has 49,226 distinct clients of which 48,586 (98.70%) are valid ages (below 100yrs) and 4,997 (10.28%) clients are long term users. The youth (18yrs to 24yrs) population in the data is made up of 5,179 (10.52%) clients. Entries are from July 1, 2009 to March 3, 2022. Each row of the table, represent a single engagement with the shelter system, such as sleep, counselling session, log of an encounter, or an occurrence that led to a ban for a period. Each interaction is time-stamped, has a unique ID, and has a distinct table index. The dataset contains 11 non-keyword count attributes and 32 keyword count attributes derived from the logs to provide relevant insights.

2.2 Description of Data Attributes

The data attributes are in 2 broad categories, either it is a keyword count or not. The data attributes that are not keyword counts are described below:

- ***ClientId*** - It is the distinct identification number of each client. Clients are given unique ID numbers once, for the life time of the data.
- ***Date*** – The date and time of each interaction is recorded under this attribute in the format; year-month-day hour : minute : second.
- ***EmployeeId*** – This is an encrypted version of the unique identification number of the employee who logged the data record.
- ***EmployeeIsCounsellor*** - An entry of this attribute is either 1 or 0 to indicate whether the employee who logged the interaction is a counsellor or not. If an employee is a counsellor, the entry is 1, otherwise the entry is 0.
- ***EMSLogFlag*** - An entry of this attribute is either 1 or 0 to indicate whether EMS have been contacted for the client or not. If EMS was contacted, the entry is 1, otherwise the entry is 0.
- ***PoliceLogFlag*** - An entry of this attribute is either 1 or 0 to indicate whether the police have been contacted for the client or not. If the police was contacted, the entry is 1, otherwise the entry is 0.

- ***BarDuration*** – This attribute is a record of the length of time the client is not permitted to access the emergency shelter. A client could be barred from accessing shelter services if the client breaks certain shelter rules or laws. The values of this attribute are strings.

- ***Location*** – The location within the emergency shelter where the clients interaction took place (e.g. room, floor, etc.) is recorded under this attribute. The values of the attribute are strings.

- ***EntryType*** – The purpose of the interaction which could be Bar, CounsellorsNotes, ProgressDetails, Log, or Sleep is recorded under this attribute.

- ***ClientState*** – The client may be sober, intoxicated, drugged, under, or drugged & intoxicated at the time of interaction with the shelter. This state of the client is recorded under this attribute.

- ***Age*** - This is the age of the client at the time of the event.

The keyword count attributes are counts of the number of times keywords related to each category occurs in the comments field related to an interaction in the DI database. A one in the health column, for example, indicates that a health-related keyword was identified in the entry's comments field. It does not, however, imply that the client has a medical condition. Keyword count attributes have values that are greater than or equal to zero. These keywords count categories include; addiction, bar, biometrics, brawl, CPS(Calgary police service), conflict, death, EMS (Emergency medical service), education, employment, financial, friends and family, gun, health, housing, ID, indigenous, justice, knife, medication, mental health, negative word, overdose, physical health, physical violence, positive word, property, seniors, sexual violence, spray, supports and weapon.

2.3 Data Preprocessing and Transformation

Real-world data is messy. It is necessary to ensure that it is in a suitable format for the training model to learn from. Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and an important step in the process of creating a machine learning model. When data is presented in a format that highlights the key components necessary to solve an issue, machine learning algorithms perform best. Transforming the data by organising different aspects of the data to make the most sense for the goal was a very important part of the data preparation. This includes combining salient variables when it makes sense like (keyword counts) and identifying important ranges to focus on (i.e., adult, youth, observation windows, housing ready, housing first and covid19).

In most machine learning methods, each ”example” is represented in the training dataset by a row, with each column displaying a different feature of the ”example”. The concept of an ”example” is discussed in Section 3.1. Data in this format is consistent and organised in a way that allows machine learning algorithms to easily analyse it. This data format has various advantages:

1. **Ease of Manipulation:** The data is easier to manipulate and analyse, allowing data scientists to focus on the analysis rather than the structure of the data.

2. **Simplified Visualization:** It facilitates better comprehension and transmission of the data insights by making data visualisation easier and more intuitive.

3. **Better ML Model Performance:** Machine learning algorithms perform better with this type of data because they can learn more effectively from consistent, accurate information.

The dataset for this study does not provide these additional benefits because a client with several interactions with the DI centre is represented in multiple rows of the data. A time window feature summation strategy was utilised to provide structured data that reflects the problem being solved as input to the ML algorithms.

2.3.1 Cohorts

For the duration of the data there have been significant policy changes and events that may have an impact on the clients' interactions with the DI centre. The approach to homelessness between 2009 and 2017 was referred to as “housing readiness”. This concept held that, people experiencing homelessness needed to be ”ready” to comply with certain standards such as sobriety or abstinence before moving out of shelters and into transitional and permanent housing [42]. In 2017, the DI centre made a significant change in its approach to fighting homelessness. This new strategy is based on the ”housing first” concept. Housing first places people experiencing homelessness into independent and permanent housing as soon as feasible, with no preconditions, and then providing them with any extra services and supports that are required [43]. The core idea of housing first is that persons who are first housed are more likely to succeed in their life. Between 2019 and 2022, the outbreak of covid19 changed a lot of things and this needs to be considered. These policy changes and events within the life of the data may have impacted the data differently. It is therefore important that we treat the data differently. The data was processed into three broad cohorts:

- **Housing ready** => from July 1, 2009 to July 31, 2017

- **Housing first** => from August 1, 2017 to February 28, 2020

- **Covid19** => from March 1, 2020 to March 31, 2022

The major part of this work is focused on the youth but at the same time to some extent the adult population is investigated as well. For this work the age bracket considered as youth is from 18 years to just under 25 years while 25 years to 100 years are considered as adults. The data was then further segregated into youths and adults to efficiently investigate each group separately.

2.3.2 Data Labelling

The problem of homelessness has been discussed into details in Section 1.1. As mentioned in Sections 1.1 and 1.2, several efforts are being made to address this problem. Some of these measures are aimed at providing stable housing and support services for individuals living with chronic homelessness. Other measures are aimed at preventing the occurrence of chronic homelessness in the first place. To achieve this, persons living with chronic homelessness or at risk of becoming chronically homeless needs to be identified. Chronically homeless people often end up being long term shelter users, because emergency shelters are among the primary points of contact to people who lose their homes. As mentioned in Section 1.3, machine learning is a powerful tool that can predict future occurrences of an event (in this case long-term shelter users) by learning from historical data. A crucial part of the learning process is identifying who is a long-term shelter user and who is not, within the historical data. The machine can then learn useful patterns within the data, that led to this event, to be able to make predictions on new unidentified data. Consequently, labelling the input data of the supervised machine learning algorithms is an essential facet of the work.

Data labeling is the process of identifying raw data and adding one or more relevant and informative labels to offer context so that a machine learning model can learn from it. For this work, labels indicate whether a client is a long-term shelter user or not. AI and machine learning algorithms learn from labelled data. As a result, data labelling is one of the most important aspects of data preparation for ML, particularly for supervised learning. Supervised machine learning is a subcategory of machine learning and artificial intelligence that uses labelled datasets to train algorithms to reliably classify data or predict outcomes accurately. Three significant definitions were examined in order to label the data.

1. **The Government of Canada's (GoC) definition:** Chronic homelessness refers to "Individuals who are currently experiencing homelessness and who meet at least 1 of the following criteria: they have a total of at least 6 months (180 days) of homelessness over the past year or they have recurrent experiences of homelessness over the past 3 years, with a cumulative duration of at least 18 months (546 days)" [44].

2. **The Government of Alberta's (GoA) definition:** "A person or family is considered chronically homeless if they have either been continuously homeless for a year or more, or have had at least four

episodes of homelessness in the past three years. In order to be considered chronically homeless, a person must have been sleeping in a place not meant for human habitation (e.g. living on the streets) and/or in an emergency homeless shelter" [45].

3. **The Calgary Drop-In Centre's (DI) definition:** Chronic homelessness refers to "Individuals who access the Calgary Drop-In Centre more than 75% or 276 days in a calendar year" [46].

The general idea of all the above definitions is that an individual must be living with homelessness for a certain period of time (measured in days) to be considered chronically homeless. Rather than focusing on the traditional definitions of chronic homelessness, we used a much simpler definition that still achieves the same results by saying a client still interacting with shelter after T_H days is referred to as a long-term client. This simplified definition also has and added benefit of potentially improving the performance of the machine learning algorithms. For this work T_H days was set at 180 days.

The threshold of 180 days was carefully chosen since it is consistent with the above definitions and has a wide coverage that encompasses nearly all the facets of these definitions. Returning to the definitions, the GoC's definition is divided into two portions, both of which emphasize that a person must have been homeless for more than 180 days. The threshold covers this definition entirely. The DI center's definition is simple: the individual must have used the shelter for more than 276 days. Again, the threshold covers this definition entirely. The GoA's definition is divided into two sections. For the first which is covered entirely by the threshold; the individual must have been homeless for more than 365 days continuously. The second part is a bit tricky; for example, a person who had four episodes of homelessness of ten days each would have accumulated 40 days of homelessness and be classified as chronically homeless, which looks good; however, a second person who had only one episode of homelessness of 300 days would not be classified as chronically homeless, but this individual is in dire need of housing support and could have been identified based on the simplified definition's threshold. Setting the threshold at 180 days captures all chronically homeless individuals except for the minority who experience repeated but not severe homelessness.

To label the data, the total number of days from the date each client starts interacting with the emergency shelter to the dates the client ceases to interact with the emergency shelter is calculated, denoted as T_L. The function below is then used to generate the label vector (y).

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{cases} \text{True:} & T_H \leqslant T_L \\ \text{False:} & \text{Otherwise} \end{cases}$$

The label column is a boolean vector, for every example, a true is returned if T_L is greater than or equal to T_H and false returned if otherwise. The labels assigned to the $1^{\text{st}}, 2^{\text{nd}}$, ..., and the n^{th} client in the dataset are denoted as $y_1, y_2, ..., y_n$.

2.3.3 Observation Windows

The idea behind preventing long-term shelter use is to prevent someone from getting so stuck in homelessness that it becomes nearly hard for them to escape. The situation of the homeless youth population quickly gets worse on all fronts: physically, emotionally, materially and socially. Early on in their shelter journey, persons who are at danger of long-term shelter usage can be identified, to benefit from quick rehousing techniques (like the housing first program) to guarantee that people transition into safe and adequate housing with services that they require. This work's solution is a tool to supplement human decision making by identifying clients who ought to be directed for consultations with real counsellors. As discussed in Section 1.3, one of the main goals of this work is to provide a solution that offers real time program delivery for the emergency shelter. It is therefore important for the model to be able to make predictions periodically on available data to be referred for consultation with counsellors.

As described in Section 2.4, the data contains several attributes some of which are keyword counts and others are not. Among the non-keyword counts, three attributes (Client ID, Date, and Event Type) are relevant to the work at hand. These three non-keyword attributes columns are extracted. This new data frame's index is set to the Client ID attribute. The time component of the date attribute was converted to midnight. The values of the entry type attribute are categorical. A categorical attribute is one with label values. The label values in the event type attribute column are as follows: bar, counsellors notes, progress details, log, Sleep, BuildingCheckIn, Mail, Message, or CondEntry. Many machine learning algorithms cannot work directly on label values. They require that all input features be numerical. This means that categorical data must be transformed into numerical data. A technique known as one hot encoding was used to accomplish this. Separate columns are prepared for each categorical label in one hot encoding. Wherever there is a label, the value in that label's column will be 1 and 0 in the columns of the other labels. Let us illustrate with an example: Assuming Table 2.1 is clients shelter interaction data. The table has categorical values for event type. Table 2.2 shows the results of using one-hot encoding on the table.

Client ID	Date	Event Type
001	2020/01/19	Sleep
001	2020/01/21	Mail
002	2020/02/15	Mail
001	2020/02/16	Sleep
003	2020/02/27	Sleep
002	2020/03/02	Sleep
002	2020/04/17	Mail
001	2020/05/11	Mail
003	2020/05/25	Sleep

Table 2.1: Illustrative client data

Client ID	Date	Event__Sleep	Event__Mail
001	2020/01/19	1	0
001	2020/01/21	0	1
002	2020/02/15	0	1
001	2020/02/16	1	0
003	2020/02/27	1	0
002	2020/03/02	1	0
002	2020/04/17	0	1
001	2020/05/11	0	1
003	2020/05/25	1	0

Table 2.2: Illustrative one-hot encoded client data

Client ID	Date	Event__Sleep	Event__Mail	Addiction
001	2020/01/19	1	0	5
001	2020/01/21	0	1	0
002	2020/02/15	0	1	3
001	2020/02/16	1	0	0
003	2020/02/27	1	0	0
002	2020/03/02	1	0	0
002	2020/04/17	0	1	7
001	2020/05/11	0	1	10
003	2020/05/25	1	0	1

Table 2.3: Illustrative one-hot encoded client data concatenated with keyword count attribute

The keyword count attributes are numeric attributes and do not need any conversion. These keyword count attributes data are then concatenated with the one hot encoded data to create the full dataset. Table 2.3 represents concatenation of a keyword attribute (Addiction) with table 2.2.

The model should be able to predict if a client will eventually become a long-term shelter user using the client's first T_o days in shelter. To achieve this the data is processed into time windows, called observation windows (30days, 60days, 90days, 120days and 180days). The window sizes were chosen with the assumption that the emergency shelter staff meets monthly. The objective is to model the data in such a way that when algorithms learn from it, they output decision rules in DNF format that can make effective predictions every 30 days. These predictions would be done to coincide with their meetings, so that persons identified as being at risk of long-term shelter use might be considered for deliberations in the meetings.

Client ID	Event__Sleep	Event__Mail	Addiction
001	2	1	5
002	1	1	3
003	1	0	0

Table 2.4: Illustrative 30days window size data

To create the 30days observation window data, each client's interaction for the first 30days is summed up and represented as a single row. For each feature the first 30days are added together and represented as a single datum. The rows in this new data are then unique clients' total interactions with the emergency shelter for the past 30days. To illustrate this better see Tables 2.4. Client 001 started interacting with the shelter on January 19, 2020. Thirty days from this date, this client has accumulated 2 event sleep services, 1 event mail service and 5 addiction counts which are represented as a single row in the table. This process is repeated for the past 60days data, 90days data, 120days data and 180days data to generate the 60days observation window data, 90days observation window data, 120days observation window data and 180days observation window data. To give early identification a chance, rules generated on the 30days observation window are applied to a client's first 30 days data of interaction with the shelter to find out if the client is at risk of long-term shelter use or not. However, some clients may still slip through the cracks. To help reduce the possibility of clients slipping through the cracks, for clients that were not captured by the 30 days rules, the same rules can be re-applied to their last 60 days data or the rules generated from the 60days observation window can be applied to their last 60 days data to ascertain if these clients are at risk of long-term shelter use or not. Finally, clients who still slipped through, can be identified in similar fashion until 180 days. Below is a mathematical representation of how the observation windows data are created.

$$X_w \in \Re^{N \times K} \quad \text{where N = number of clients, K = number of features and w = window size}$$

$$= \begin{bmatrix} X_{1,w} \\ \vdots \\ X_{n,w} \end{bmatrix}$$

$$X_{n,w} = \begin{bmatrix} f_{1,w} & \dots & f_{k,w} \end{bmatrix}$$

Where;

$\Re$ is a matrix with N rows and K columns.

X_w is a matrix of N row vectors.

Each row vector $X_{n,w}$ contains k features.

$f_{k,w}$ is the number of times feature k occurs in the n$^{\text{th}}$ first w days in shelter.

2.4 Exploratory Data Analysis

An exploratory data analysis (EDA) was performed to analyze and summarize the data's main characteristics, employing data visualization. The EDA's major goal was to assist in looking at the data before making any assumptions. The main benefit of the EDA was a better knowledge of the variables in the dataset and their relationships. It assisted in deciding how to best alter the data to obtain the required answers. It also assisted in determining the suitability of the statistical methods under consideration. EDA methods are still often employed in the data discovery process today. Exploratory analysis is performed to make sure the outcomes are reliable and relevant to the targeted objectives. By ensuring that the correct questions are being asked, EDA is also helpful. After EDA is finished and insights drawn, its features are then used for machine learning.

2.4.1 Raw Data EDA

The data has 5,576,433 interactions with the shelter system distributed over its entire life span (2009 – 2022). It can be seen from Figure 2.1 that, there have been high interaction of clients with the shelter system between 2009 and 2016. It coincides with the period when housing readiness was the approach used by the DI center to address homelessness. As mentioned in Section 2.3.1, clients needed to be ready to comply with certain conditions before being considered for permanent housing support. As discussed in Section 1.2, long term clients are usually the minority of clients that use shelter services the most. Though several factors could contribute to this high volumes of interactions; like a lot of new people entering the shelter system but it is also possible that several clients (long term clients) were having trouble complying with these conditions which left them cycling in the emergency shelter system.

The volume of interactions started to decline slowly till 2019. During this period The DI center made a shift in their approach to addressing homelessness. The housing first philosophy was adopted. As discussed in Section 2.3.1, with the housing first approach, clients did not need to be ready or comply to any condition before being considered for permanent housing support. Long term clients were transitioned into permanent housing before other support services were provided to them to ensure a stable and decent living. This made it easy for long term clients to exit the shelter system with ease. Though it can not be concluded that this was the reason for the decline in shelter usage but it is only logical that this is one of the factors.

The volume of interactions suddenly went up and peaked in 2021. Obviously this is the covid19 pandemic period and it affected every single sector and the homeless population was one of the hard-hit group. With all the lock-downs and reduced essential services, a portion of the homeless population which mostly avoid the emergency shelters for various reasons and engage in rough sleeping suddenly came into contact with

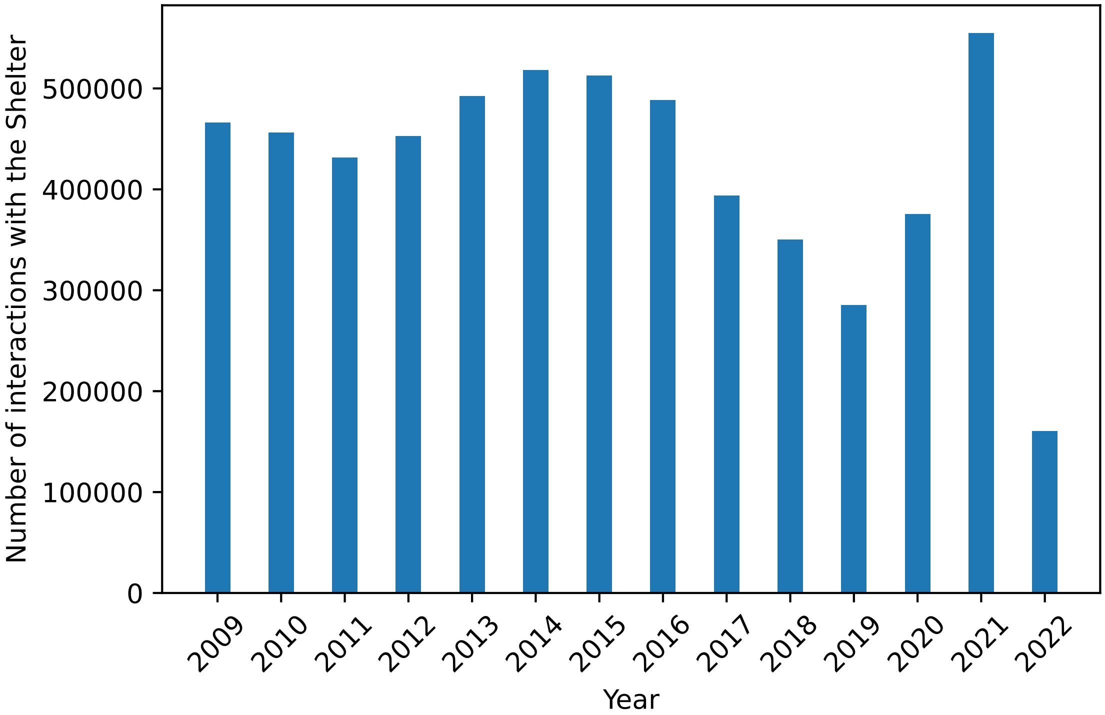

Figure 2.1: Shelter interactions distribution for the data period

the shelter system.

2.4.2 EDA on the Different Cohorts

Table 2.5 outlines the total number of clients, total number of long-term clients and also the probability of a client being a long term client in each cohort data. The cohort column shows the different cohort datasets, the client count column specifies the total number of clients in the cohort, the long-term client count column specifies the total number of long-term clients in the cohort while the probability of long-term client shows how likely it is to pick a long-term client from the cohort at random. The definition for long term clients has been discussed in section 2.3.2. The threshold a client needs to meet while still interacting with the shelter to be considered as a long-term client used to generate Table 2.5 was 180 days.

The proportion of clients labelled as long term clients in Table 2.5 is unusually higher in some of the cohorts(all adult, all youth, housing ready adult and housing ready youth) as compared to other works where the chronic clients proportion is always below 10%. The chronic clients proportion is 4.8% in [29], 8.4% in [38] and 9.9% in [39]. This work defines long term shelter users in a much simpler way but aimed at achieving the same goal to the more traditional definitions. The idea is to be able to capture most heavy users of the emergency shelter system for permanent housing support. If a client accesses sleep services for 180 days and

Cohort	Client Count	Long Term Client Count	Prob. of Long Term Client
AllAdult	24599	10985	0.447
AllYouth	5179	2467	0.476
Covid19Adult	2350	497	0.212
Covid19Youth	377	84	0.223
HousingReadyAdult	12135	3703	0.305
HousingReadyYouth	2688	859	0.320
HousingFirstAdult	3580	386	0.103
HousingFirstYouth	652	71	0.109

Table 2.5: Statistics of the cohorts

is still interacting with the shelter system for sleep services, the client definitely needs help. For this work the definition of long term clients applies to the life span of a clients records in the data other than the client having heavy usage of shelter within a one year period or a recurrent episodic usage of shelter within a three year period. This accounts for the higher proportions of clients being classified as long term clients in the table.

Additionally, the significant proportion of clients classified as long-term shelter users in the all adult cohort and all youth cohort is particularly advantageous to the machine learning models. This is because the models have a sufficient number of instances from both classes to learn from, since the data is closer to being balanced. When the number of chronic clients in the data is very low (e.g. the housing first adult cohort and housing first youth cohort), additional strategies are used to counteract the data's imbalance, such as sampling to balance the data or assigning weights to the various classes to penalise the models more severely for misclassifying the minority class. A model might get biased towards particular predictions depending on the data it encounters during training, producing a high percentage of accurate predictions but subpar overall performance. When a model is trained on an unbalanced dataset, it may appear to have very accurate predictions, but this accuracy is misleading. Let's imagine for example, that a data has 5 percent ”long-term clients” whereas the rest of the data is made up primarily of ”not long-term clients.” During training, the classification model discovers that it can predict ”not long-term clients” for every piece of data it sees with a 95 percent accuracy rate. This is a major issue because the model is supposed to identify the ”long-term clients.”

2.4.3 Statistical Characteristics of the Transitional and Long Term Clients

Table 2.7 outlines the statistics of the data features. A detailed description of the data attributes has been provided in Section 2.2. As can be seen in the table, some of the keyword count attributes are not shown in the table. This is because similar keyword count attributes were merged into single features. In general, the keyword counts are not powerful discriminators. This is reflected in the decision rules presented in Section 5.7. Having several correlated features in a dataset does not improve a machine learning model. Due to the curse of dimensionality, fewer features usually result in a faster learning by the algorithms. It is thus very beneficial to combine correlated keyword count data features. The keyword count attribute combinations are shown below;

- The **overdose** and **addiction** attributes were combined to create the **addiction feature**.
- The **conflict, physical violence, brawl, knife, gun, spray, weapon,** and **sexual violence** attributes were combined to create the **conflict feature**.
- The **CPS** and **justice** attributes were combined to create the **police justice feature**.
- The **supports, housing, financial, employment** and **education** attributes were combined to create the supports feature

Some non keyword attributes such as client ID, date, employee ID, employee is counsellor, location, client state and age are not correlated with the target label hence are dropped. The correlation between the target label and these non keyword attributes are shown in Table 2.6 to support the above assessment.

Non-Keyword Attribute	Correlation
ClientId	-0.129504
Date	-0.018257
EmployeeId	-0.032454
EmployeeIsCounsellor	-0.037543
ClientState	0.067724
Age	0.118168
Location	-0.007520

Table 2.6: Correlation between the label and the dropped non keyword attributes

We attempt to get a preliminary insight of how different the long term clients are interacting with the shelter, compared to the not-long term clients. Table 2.7 shows that, for a majority of the data features, there is a wide disparity between the long term and the not-long term clients' experiences. It can be noticed that sleep services is by far the most accessed emergency shelter service by both long-term and not long-term clients despite the fact that long-term clients access the service more heavily. Other features recording a wide disparity of interactions as shown in the table are counsellors notes, conflict, supports, log and addiction.

These features recording wide disproportion of interaction between the long-term and not long-term clients could be the most useful features in predicting future events. It can be seen that features like mail, EMS, message and cond entry are recording very low interactions.

	Minimum		**Maximum**		**Median**		**Mean**		**Zeros%**		**95th Percentile**	
Features	Not-LT	LT	Not-LT	LT	Not-LT	LT	Not-LT	LT	Not-LT	LT	Not-LT	LT
Addiction	0	0	64	466	0	1	0.6	9.9	85.3	48.0	4	49
MentalHealth	0	0	51	401	0	0	0.3	2.9	91.2	63.9	1	14
PoliceJustice	0	0	204	182	0	0	0.4	4.4	86.1	52.6	2	22
EMS	0	0	50	191	0	0	0.1	1.2	97.4	80.5	0	7
Supports	0	0	274	948	0	5	2.8	24.2	55.8	25.4	13	22
Conflict	0	0	284	2023	0	2	1.3	29.7	82.9	41.6	5	151
Bar	0	0	20	165	0	0	0.1	2.4	95.9	63.6	0	12
CounsellorsNotes	0	0	95	598	0	1	0.5	5.3	76.5	45.0	2	26
ProgressDetails	0	0	60	539	0	0	0.4	6.1	83.4	50.8	2	32
Log	0	0	176	2419	0	2	1.1	15.5	53.6	27.6	4	67
Sleep	0	0	244	4757	1	20	5.6	150.4	35.8	6.9	29	763
BuildingCheckIn	0	0	420	1813	0	0	2	26.0	90.6	75.4	3	149
Mail	0	0	7	73	0	0	0	0.6	99.2	89.9	0	3
Message	0	0	11	37	0	0	0	0.2	98.8	90.5	0	1
CondEntry	0	0	6	47	0	0	0	0.7	96.6	78.0	0	4

Table 2.7: Event statistics of not-long term and long term clients

Table 2.8, illustrates the shelter access data for long-term and non-long-term guests for the various cohorts. The median, mean, and 90th percentiles of total stays, episodes, and days are displayed. A stay is defined as a 24-hour period during which sleep services are accessed at least once. Episodes are defined as shelter stays separated by periods of 30 days or longer with no record of having used shelter [47]. Days are the number of days between the first and last stay of a client.

Cohort		Stays			Episodes			Days		
		Med	Mean	90th %tile	Med	Mean	90th %tile	Med	Mean	90th %tile
HousingReadyYouth	LT	9.0	41.32	107.00	3.0	3.52	6.00	707.0	916.80	1910.00
	Not-LT	1.0	4.89	10.00	1.0	1.12	2.00	1.0	17.30	61.00
HousingReadyAdult	LT	7.0	64.96	176.00	3.0	3.56	6.00	798.0	956.90	1905.00
	Not-LT	1.0	5.10	9.00	1.0	1.12	2.00	1.0	17.45	65.00
HousingFirstYouth	LT	10.0	27.98	74.00	2.0	2.63	4.00	386.0	411.18	662.00
	Not-LT	1.0	4.09	7.00	1.0	1.12	1.00	1.0	15.87	50.00
HousingFirstAdult	LT	9.0	37.10	113.00	2.0	2.80	4.00	383.0	410.01	659.00
	Not-LT	1.0	4.21	7.00	1.0	1.11	1.00	1.0	14.78	53.00
Covid19Youth	LT	41.0	80.33	220.00	3.0	3.03	5.00	361.0	372.35	593.00
	Not-LT	3.0	11.25	31.00,	1.0	1.29	2.00	8.0	38.34	123.00
Covid19Adult	LT	62.0	108.95	271.00	3.0	2.84	5.00	356.0	376.69	602.00
	Not-LT	2.0	12.49	36.00	1.0	1.15	2.00	5.0	29.65	111.00
AllYouth	LT	19.0	73.75	194.00	4.0	5.12	10.00	1456.0	1665.79	3369.00
	Not-LT	1.0	5.45	12.00	1.0	1.14	2.00	1.0	19.51	74.00
AllAdult	LT	24.0	147.37	411.00	4.0	5.00	10.00	1320.0	1570.11	3211.00
	Not-LT	1.0	6.02	13.00	1.0	1.13	2.00	1.0	19.09	75.00

Table 2.8: Shelter access statistics

2.5 Summary

The data used for this work is an anonymized records of clients interaction with the DI center. The data attributes are either keyword counts or non-keyword counts. The data was processed into housing ready, housing first and Covid19 cohorts before further segregated into youths and adults to efficiently investigate each group separately. To label the data, definitions of chronic homelessness from the Government of Canada, the Government of Alberta, and the Calgary Drop-In Centre were analysed, and a far simpler definition (a client still interacting with shelter after TH days is referred to as a long-term client) was arrived at. The data was processed into observation windows (30days, 60days, 90days, 120days, and 180days) so that the model could generate predictions on a regular basis. Categorical data was converted to numerical data using one hot encoding. An EDA was also performed on the raw data, displaying the clients' shelter access pattern over the data's time span. EDA on the various cohorts revealed that the proportion of clients labelled as long term clients is unusually higher in some of the cohorts (all adult, all youth, housing ready adult, and housing ready youth) when compared to other studies where the number of chronic clients is always less than 10%.

Chapter 3

Concept Review

All the ML models considered in this work fall within a group called logical models [48]. The reasoning behind the choice of logical models for this work has been clearly explained in Setion 1.5. Logical models divide the data into segments using logical expressions and thus build groups. The goal is to find a segmentation that makes the data in each segment more homogeneous in relation to the task at hand. For example, in classification, a segmentation in which the examples in each segment are predominantly of one class is sought. There are two types of logical models: tree models and rule models. Rule models are further subdivided into rule list models and rule set models. It is worth mentioning that rule list models were not considered for this work due to their close relation to tree models. A rule list is basically a one-sided tree. Logical models are generally similar; they consist of a collection of rules, though in tree models the rules are organised in a tree structure.

The building blocks and the structure of logical expressions of the different logical models and how they can be translated into a unified form is described in this chapter. The rest of this chapter is organised as follows; Section 3.1, describes the unit building block (thus a literal) of all logical models. In Section 3.2, a "term" of logical models is explained. Section 3.3 details the structure of tree models and how a tree is induced. A "Rule" of a logical model is discussed in Section 3.4. In Section 3.5, the technique employed to translate the tree to a DNF format is described. Finally, the summary of the chapter is presented in Section 3.6.

3.1 Literal

Let's briefly explain the terms ”features” and ”examples” which are repeatedly used throughout this literature before delving into the concept of literal. “Features” are the independent variables or attributes of the data. For instance, the ”features” in Table 3.1 are humidity, weather and wind. An “example” is an observation represented by a pair (x, y), where x is a vector of features and y is a label. In simple terms an “example” is a particular instance of the data. An “example” is unique and distinguishable from other “examples”. For Table 3.1 the “examples” are the weeks. Putting this all together, the rows of the data represent all the different weeks that the data has (each row is a specific week thus; an example) and the columns represent the features.

Week	Humidity	Weather	Wind	Label
001	24	Sunny	Strong	No
002	30	Cloudy	Weak	Yes
003	36	Sunny	Weak	Yes
004	36	Sunny	Strong	No
005	42	Sunny	Strong	No
006	44	Cloudy	Weak	Yes
007	46	Cloudy	Strong	No
008	47	Cloudy	Weak	Yes
009	47	Sunny	Weak	No
010	51	Cloudy	Weak	Yes

Table 3.1: Meteorology data

Equalities of the form “feature = value” and, for continuous features, inequalities of the form “feature $\leqslant$ value” are the simplest logical expressions; these are known as literals. A literal 'l' is a feature-operator-value triad, l = (f, o, v), in which 'f' is a feature, 'o' is a relational operator ($\leqslant, =, >$), and 'v' is a value [26]. This is the basic building block of logical models. The literal takes a single feature of an example and compares it with the threshold value, then returns a boolean (True or False). The goal of a literal is to divide the data into homogeneous groups. The more bias it is towards the target class the better it is. If an example meets the conditions of the literal, the literal is said to have captured the example. For instance a literal:

$$\mathrm{l} = (\mathrm{Humidity} > 43)$$

applied to Table 3.1 captures 3 ”Yes” & 2 ”No”. This literal is 60% precise at predicting the ”Yes” class. A very efficient way to improve performance is by combining multiple literals together using logical connectives (e.g. conjunction ”$\wedge$”, IF....THEN, etc.) to benefit from the relations multiple features have with the class labels. The concept of combining multiple literals together for better performance has been discussed in the next section (3.2).

3.2 Term

Logical connectives (e.g. conjunction ”∧”, implication ”IF....THEN”, etc.) can be used to construct more intricate boolean expressions that can segregate the examples better by leveraging multiple features of the data. In rule based models, a term ’t’ is a conjunction of literals, whereas tree models use nested implications to form a term. By combining another literal l = (Weather = Cloudy) with the literal discussed in Section 3.1, results in a term. A rule based model’s term will be represented as:

$$\text{t} = (\text{Humidity} > 43 \ \wedge \ \text{Weather} = \text{Cloudy})$$

while a tree model’s equivalent of the same term will be represented as:

IF Humidity > 43
↳ **IF** Weather = Cloudy
↳ **THEN** Predict ”Yes”

Three ”Yes” and one ”No” are captured by both terms (rule based and tree term). The terms are 75% precise at predicting the ”Yes” class, which is an improvement to the single literal. The output of both terms is the same but how they arrive at the output is different. The tree model’s term is ordered, meaning the first literal must be applied to the example first, if the example meets that condition then the second literal is applied to the example again. On the other hand the rule based model’s term is unordered, hence any of the literals can be applied to the example first or second.

A single term could do a good job of classifying the examples of a very simple dataset with few features and feature-values that are strongly correlated with the labels. In real world problems the dataset is mostly large with a lot of features and feature-values. For a single term to be able to sufficiently discriminate between the examples according to the labels, then a few of the data features must be highly discriminatory, which is mostly not the case. Tree models and rule models induces terms differently, which is discussed in Section 3.3 and 3.4.

3.3 Tree

Figure 3.1 represents a simple tree induced to classify the data of Table 3.1. A tree’s structure is hierarchical and flowchart-like, with a root node, branches, internal nodes, and leaf nodes. From the figure, (Wind = Weak) is the root node, (Weather = Cloudy) and (Humidity < 41.5) are the internal nodes, the

” Yes” and ”No” labels are the leaf nodes. The tree has 2 branches thus, from the (Wind = Weak) to the leftmost ”Yes” is one branch and from the (Wind = Weak) to the other ”Yes” is the other branch. Nodes are feature evaluation points of the tree. The root node is the node that starts the tree and it evaluates the feature that best splits the data. Leaf nodes are the final nodes of the tree, where the predictions are made. Internal nodes are nodes where additional features are evaluated but which are not the final nodes where predictions are made. The path from the root node to a single positive leaf node is called a branch of the tree. The tree grows from a single node (the root node), with no incoming branches. The root node's outgoing branches then feed into internal nodes, also known as decision nodes. Both node types undertake evaluations based on the available attributes to create homogeneous subsets, which are denoted by leaf nodes or terminal nodes.

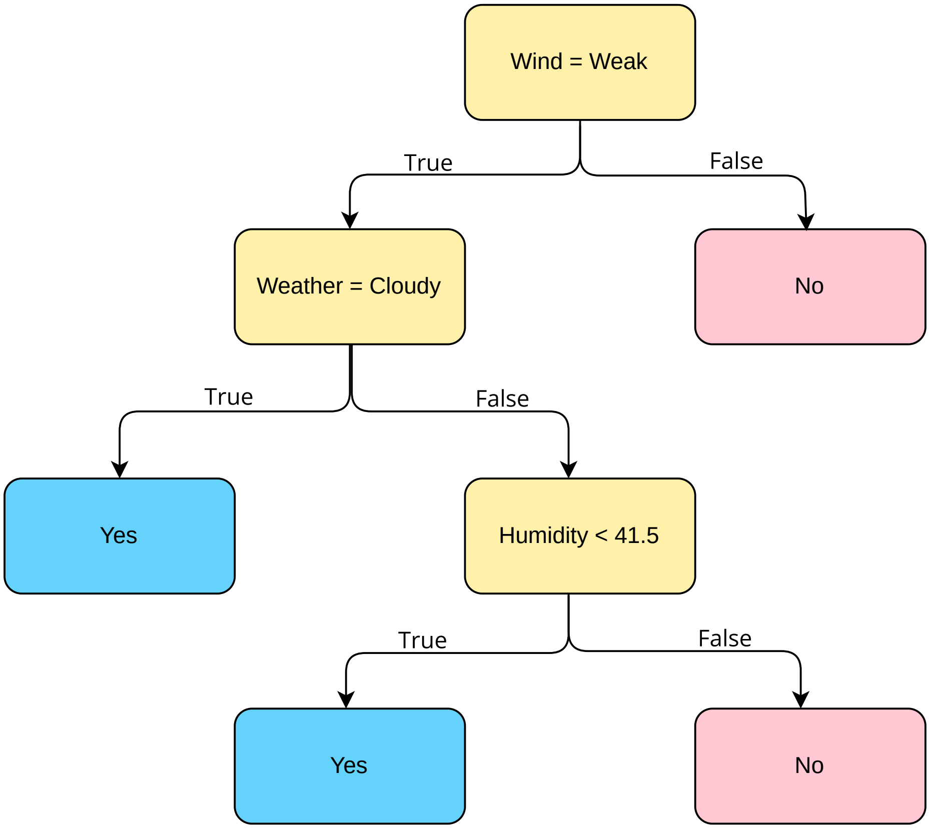

Figure 3.1: Decision tree from meteorology data

A leaf node represents the classification assigned examples that it covers. For instance, to grow the CART [49] tree, the learning process uses a divide and conquer strategy by undertaking a greedy search to discover the optimal split points (nodes) within a tree. The nodes are added incrementally, using labelled training examples to guide the choice of nodes. The training examples are split into ever-smaller subsets continually, resulting in a tree. Each internal node has children equal to the number of outcomes to its condition.

Unifying the concept of tree with the concepts presented in Sections 3.1 and 3.2 provides a clearer understanding of this literature. A node of a tree is basically a feature evaluation which is a direct equivalent of a literal. All the conditions in the branch of a tree must be met for an example to be classified by the leaf of that branch, therefore it is an ordered conjunction of all the literals within it. The representation of a single branch is therefore a term. In this work a literal will be used for a node and a term will be used for a branch.

The step by step process to induce the tree in Figure 3.1 which predicts whether it will rain in a particular week or not, using the simplified data in Table 3.1 is presented. The most important step in creating a decision tree, is the splitting of the data (D) into subsets (D_1) and (D_2). There are different criteria that can be used in order to find the next split. Gini Impurity, which is the criterion used by CART algorithm will be the focus.

The Gini Impurity for a data set D is calculated as follows:

$$\text{Gini Impurity}(D) = \frac{n_1}{n} \times \text{Gini}(D_1) + \frac{n_2}{n} \times \text{Gini}(D_2)$$

where $n = n_1 + n_2$ the size of the data set (D) and

$$\text{Gini}(D_i) = 1 - \sum_{j=1}^{c} p_j^2$$

where: The particular subset of the data created by the literal is denoted "i", e.g. D_1 and D_2 for binary classification. The total number of different classes in the data is denoted as c. For Table 3.1, c = 2. The probability of samples belonging to class j at a given literal is denoted as p_j. The lower the gini impurity, the higher the homogeneity of the examples captured by the literal. The gini impurity of a pure literal is zero.

To split a decision tree using gini impurity, the following steps need to be performed.

1. For each possible split, calculate the gini impurity of each child literal

2. Calculate the gini impurity of each split as the weighted average gini impurity of child literals

3. Select the split with the lowest value of gini impurity

Steps 1–3 are repeated until no further split is possible. It is worth noting that continuous features takes more calculations. The values are sorted, the mean of neighbouring values calculated and then gini impurity is calculated for each of these means.

The first step of the tree induction process is deciding the feature to use for the root literal, which is the most discriminatory literal. The target class is predicted with only one feature at a time and then the feature that has the lowest gini impurity is used as the root literal. The gini impurity for all the possible splits are shown in Table 3.2. It can be seen that the lowest gini impurity is when using wind, hence this is the root literal and the first split. The root literal divides the data into 2 subsets. One subset to the right and the other to the left. The literal to the right of the root (captured 0 ”Yes” and 4 ”No”) is already pure, therefore this literal is turned to a leaf, and no further splitting is necessary. The literal to the left of the root (captured 5 ”Yes” and 1 ”No”) is not pure and can be split further.

Split	Gini Impurity
Humidity < 27	0.441
Humidity < 33	0.5
Humidity < 36	0.5
Humidity < 39	0.5
Humidity < 43	0.48
Humidity < 45	0.5
Humidity < 46.5	0.475
Humidity < 47	0.475
Humidity < 49	0.441
Weather = Cloudy	0.32
Wind = Weak	0.168

Table 3.2: Gini Impurity to identify the root node

The subset of data to the left of the root literal is represented in Table 3.3. To further segregate this remaining data, the same process is repeated by calculating the gini impurity for each of the remaining features (humidity and weather) on the examples. The gini impurity for all the next possible splits are shown in Table 3.4. It can be seen that the lowest gini impurity is the split, weather = cloudy, hence the next split. The subset of data to the left of this literal is pure (4 ”Yes” and 0 ”No” captured) and needs no further splitting. The subset of data to the right of this literal is not pure (1 ”Yes” and 1 ”No” captured) and can be split further. Only one feature (Humidity) with two values remain, the mean value is one value, hence the final split is made on that.

Week	Humidity	Weather	Label
002	30	Cloudy	Yes
003	36	Sunny	Yes
006	44	Cloudy	Yes
008	47	Cloudy	Yes
009	47	Sunny	No
010	51	Cloudy	Yes

Table 3.3: The subset of meteorology data for the next split

Split	Gini Impurity
Humidity < 33	0.26
Humidity < 40	0.25
Humidity < 45.5	0.22
Humidity < 47	0.22
Humidity < 49	0.26
Weather = Cloudy	0.16

Table 3.4: Gini Impurity to identify the next split

The complexity of the decision tree plays a significant role in determining the extent of homogeneity of data points captured by the leaves. However, when a tree gets bigger, it gets harder to understand and typically leads to too little data falling into a particular subtree (term). For instance, assume an internal literal "A" captures 67 positive examples and 5 negative examples. The literal further splits into two leaves, leaf "B" (capturing 2 positives and 0 negatives) and leaf "C" (65 positives and 5 negatives). Leaf "B" is pure (which is an ideal leaf) but, does capturing only two examples worth the extra complexity added to the tree? It would be preferable to stop the split at literal "A". Decision trees shouldn't be overly complex; often, the simplest explanation is the preferred one when interpretability is vital.

Pre-pruning, post-pruning, or a combination of the two are frequently used to minimise complexity. With pre-pruning, the tree's growth is constrained by a variety of criteria (such as the tree's height, the number of leaves, the minimum number of samples captured by a leaf, etc.). Post-pruning allows the tree to grow to full size and then the branches that split on features with low importance are removed.

Decision trees are very helpful for jobs involving knowledge discovery and data mining. Technical and non-technical parties can both comprehend why a decision was taken due to their boolean logic and visual structure that creates an easy to digest picture of decision-making. A decision tree's hierarchical structure also makes it simple to identify which attributes are most relevant. Different data types, such as discrete or continuous values, can be handled by decision trees.

3.4 Rule

The more literals you add conjunctively the more and more precise the term becomes at capturing the target class at the expense of the number of positive examples captured. For example, if a dataset contains 475 examples belonging to the negative class and 387 belonging to the positive class. A term that captures 58 positive examples as positive and the remaining examples as negative will score a 100% precision, but this term will let go the majority of the positive class. Learning this term alone would be detrimental. The lack of expressiveness of a single conjunctive term is the main shortcoming for learning a single term. A model can be enriched by a combination of different terms that captures different sections of the data. To achieve this the terms are combined disjunctively. Hence, the predictive model is often composed of disjunctions of terms.

“Rule” sometimes referred to as a “Rule set” refers to disjunctions of terms. This format of rule representation is referred to as disjunctive normal form (DNF). DNF (disjunctive normal form) is sometimes referred to as “disjunctions of conjunctions”, or “classification rules” but for this work we will adopt the former. Disjunctive normal form (DNF) is a sequence of ORs (disjunctions) consisting of one or more terms each of which is a conjunction (AND) of one or more literals. A rule in DNF form is considered to be in full disjunctive normal form if each of its features appears exactly once in each conjunction. Lets disjunctively combine another term t = (Wind = Weak) with the term in Section 3.2. This will result in the rule:

$$r = \{(\text{Humidity} > 43 \ \wedge \ \text{Weather} = \text{Cloudy}) \ \vee \ (\text{Wind} = \text{Weak})\}$$

This rule captures 5 ”Yes” and 1 ”No”, thereby a significant improvement over the term in Section 3.2. It does not only improve precision from 75% to 83.3% but also increased the number of positive examples captured from 3 to 5. This extent of improvement cannot be achieved by a term since a term only purifies the predictions without expanding the number of positive predictions.

For a prediction to be made by the model, an example is tested with the rule. If the example satisfies the condition of one or more terms of the rule, the example is classified as positive, else it is classified as negative. The beauty of rules in disjunctive normal form is that they are unordered. The order of the terms in the rule is of no significance. An example needs to meet the condition of any one or more terms of the rule to be classified as positive. This unordered nature of terms in the rules make them very easy to comprehend and most importantly easy to implement in a simple spreadsheet or any database. Any logical statement can be expressed using disjunctions of literal conjunctions. It is important to note that the rule could be made up of just one term. For uniformity and fairness in the assessment of models the rules of all the models considered in this work are represented or translated to DNF format.

Rule complexity ’R’ is the total number of literals in the rule. The importance of rule complexity comes to bare when sparsity is considered during rule generation. When trees are translated into DNF format, they can rapidly before very complex with increasing height. This necessitates the need to simplify the rules without altering the conditions of the rule. Minimising rule complexity is an open problem researchers are trying to solve. For instance, [50] has categorised this problem as NP-Complete and focused on finding the shortest DNF formula consistent with a truth table, where the size of a DNF formula is defined to be the number of literals in it. This study [51], investigates the problem of determining the smallest DNF formula for a function given its truth table. In 2001, [52] proved that minimum DNF is SP2-complete.

3.5 Tree Translation to DNF Rules

Rules in DNF form is the ultimate requirement for this work and has been demonstrated in Section 1.5. It is therefore incumbent to translate models that are generated by tree algorithms into DNF format. Trees are sometimes expressed in the form ”IF. . . THEN. . . ELSE. . . ”. For instance, the tree in Figure 3.1 can be expressed as:

IF Wind = Weak

↳ **IF** Weather = Cloudy **THEN** Predict ”Yes”

↳**ELSE IF** Humidity < 41.5 **THEN** Predict ”Yes”

Rules in DNF form predicts the positive class and all examples that are not captured are classified negative. Tree models generates rules for predicting the positive class and rules for predicting the negative class as well. Since complexity is a major component of the models' assessment, addressing these “duplicate rules” will enable the tree models not to be judged unfairly.

To translate a tree into a DNF format, the path from the root literal to each positive leaf is taken as a term. This is done by conjunctively combining all the literals from the root to the positive leaf. For example in Figure 3.1 the leftmost term will be:

$$\mathrm{t} = [(\text{Wind} = \text{Weak}) \wedge (\text{Weather} = \text{Cloudy})]$$

The paths to the negative leaves are discarded because all examples that are not covered by the positive description are classified as negative. All the derived terms are then disjunctively combined together as a rule in disjunctive normal form. The full decomposition of the tree in 3.1 into disjunctive normal form will then be:

$$\mathrm{r} = \{[(\text{Wind} = \text{Weak}) \wedge (\text{Weather} = \text{Cloudy})] \vee [(\text{Wind} = \text{Weak}) \wedge (\text{Weather} = \text{Sunny}) \wedge (\text{Humidity} < 41.5)]\}$$

The approach taken to translate the tree rules to DNF form is in four steps:

- First of all, rules that predict the negative class are pruned.
- In tree models, it is common to see two child literals from the same parent literal making the same decision. This added complexity does not improve the model's performance any further. To be fair with tree models, all child literals originating from the same parent literals that makes the same decisions are recursively pruned till there exist no two child literals of the same parent making the same decision.
- All the literals of each term are then conjunctively combined together.
- After which all the terms are disjunctively combined together to form the rule in disjunctive normal form.

3.6 Summary

This chapter highlighted fundamental concepts that aid in the comprehension of the literature. The building blocks (literal, term, rule) and structure (DNF, tree) of the different logical models have been explained. The literal, which has the form ”feature = value” or ”feature $\leqslant$ value,” is the fundamental building block of a logical model. A term is an improvement on the literal that is formed by conjunctively combining multiple literals to create one term that is more precise than each of the literals contained within it. A term, although very precise, lacks expressiveness, giving birth to a rule that is both precise and expressive. A rule disjunctively combines multiple terms, allowing it to capture all examples captured by each term contained in it. Following the four processes outlined in Section 3.5, tree models are finally translated to this format of rule representation known as disjunctive normal form (DNF).

Chapter 4

Algorithms Considered

It can be challenging to select machine learning algorithms. For classification problems there are a number of intriguing techniques, including decision trees, neural networks, support vector machines, and others. Each algorithm has benefits and drawbacks, and the selection must be in line with the work's goals. Rules in DNF form are the main prerequisite for algorithms for this research, as was made explicit in Section 1.5. Only algorithms that ultimately produces DNF rules were investigated. In this chapter the algorithms considered are discussed. Section 4.1 discusses the CART algorithm. In section 4.2 the GOSDT algorithm is discussed. The EXPLORE algorithm is discussed in section 4.3. Finally, the Section 4.4 recaps the discussions made in this Chapter.

4.1 CART

The Classification and Regression Trees (CART) is a well-established algorithm. A number of machine learning algorithms have specifically become popular because of their interpretability. One of the most popular types of interpretable models is the decision tree. The CART algorithm [39] has been widely utilised as a foundational model for interpretability ever since it was created. Despite several decades of efforts to increase the optimality of decision tree algorithms, CART decision tree algorithm has remained the dominant method in practise [53]. Classification and regression trees (CART) is a versatile method for determining the conditional distribution of samples given a vector of predictor values. The predictor space is recursively divided into subsets by the model using a binary tree, where the distribution of samples becomes progressively more homogeneous. The partition is determined by splitting conditions connected to each of the internal literal, and the leaves of the tree correspond to the various parts of the partition. Each observation is then allocated to a specific leaf in the tree by travelling from the root literal through to the terminal literal (leaf), where the conditional distribution of the example is found. Section 3.3 describes a thorough step-by-step approach used by a decision tree algorithm to generate a tree for a simplified sample data classification.

Complexity of the model is of significant importance in this work. CART provides ways of controlling the model complexity. These ways include:

- Limiting the tree height. Maximum tree depth is used to stop further splitting of literals when the specified tree depth has been reached during the building of the initial decision tree.
- Limiting the number of samples a leaf can capture. The most common halting strategy is to use a small quantity of training data assigned to each leaf . If the count is less than the set threshold, the split is refused, and the literal becomes a leaf.

It will however be shown later, in chapter 5 that these complexity control methods are not very effective as the models are still complex when translated to DNF format.

CART handles continuous features with ease. At every split decision point, CART performs a simple threshold test with all the means of every two consecutive values. The mean value that produces the segregation with the highest homogeneity is use at that point for the split. This threshold test is shown in the tree induction process spelt out in Section 3.3. The CART algorithm requires no specific data preparation.

CART gives an interesting and often illuminating way of looking at data in classification problem. The tree procedure output gives easily understood and interpreted information regarding the predictive structure of the data. This is clear in Figure 3.1, which epitomises a simply CART tree for classification task. The terms generated by CART are of the form "IF. . . THEN. . . ELSE. . . " presented graphically. These are ordered terms, thus the order of the literals in each term are of significant importance. Though the output model of CART are not rules in DNF form, they can easily be translated to DNF rules. Section 3.5 provides the technique employed to translate the tree to DNF rules.

4.2 GOSDT

Most decision tree models are designed to optimise on only accuracy hence are inherently unable to deal with imbalanced dataset. Accuracy could be misleading when dealing with highly imbalanced datasets. For instance, if a dataset contains 475 examples belonging to the negative class and 25 belonging to the positive class. A classifier that captures all examples as negative class without learning anything from that dataset will still be scored 95% accurate, but this classifier is totally useless. GOSDT (Generalised and Scalable Optimal Sparse Decision Trees) [24], is a novel algorithm that generates optimal decision trees for a range of objectives such as F-score, AUC, and partial area under the ROC convex hull. Even when the dataset is balanced, the ability to optimise on many metrics is still significantly useful, because some metrics are more important in particular application areas.

GOSDT offers two methods for controlling the complexity of the output model.

- Regularization: It is a decimal value within the range [0,1], used to penalize complexity. A bigger regularization value penalizes the model the most, leading to a sparser tree. On the other hand a small regularization, generates a more complex tree and could also lead to a longer training time. The complexity penalty is added to the risk in the following way.

$$\text{Complexity Penalty} = \text{Number of Leaves} \times \text{Regularization}$$

- Prunning: It is a method which penalizes the length of the tree as discussed in Section 3.3. GOSDT employs pre-prunning technique for complexity control. **Depth_budget**, which is an integer value $\geq$ 1, used to set the maximum tree depth (where 1 is just the root node and 0 means unlimited). This is a pre-pruning method used to stop the tree from growing further once the tree reaches the set maximum height enabling a less complex tree to be generated.

As mentioned earlier continuous variables make up the feature space of the dataset for our research. Continuous variables present difficulties for optimality since they increase the number of splits by the number of possible values of that variable across the entire dataset, and each extra split causes the size of the optimisation issue to grow exponentially. Most decision trees use different approaches to convert continuous variables into categorical features, sacrificing optimality. OSDT [53] and CORELS [25], for example, use a binarization approach in which each numerical property is turned into several binary features using threshold values. The bucketization approach is used in DL8.5 [54]. Using a set of thresholds, this method converts numerical attributes into categorical features by replacing all of the values inside a small interval with a single representative value for that interval. Bucketization threshold sets are determined by either distance or frequency. GOSDT is a scalable approach for generating optimal trees in the presence of continuous features while preserving optimality. GOSDT establishes a split point at the mean value between each ordered pair of distinct values present in the training data for each continuous-valued feature. However, this expands the search space, which can be computationally costly. GOSDT uses a method known as ”similar support” bounds to reduce the search space. According to [24], similar support bound states that, if two features in the dataset are similar but not identical to one another, then bounds for a split in a tree that were obtained using the first feature may be used to produce bounds for the same tree if the second feature were to take the place of the first feature. Splits at values close to "v" can utilise previous computation to quickly generate tight bounds if a split at value "v" has already been visited.

4.3 EXPLORE

The Exhaustive Procedure for LOgic-Rule Extraction (EXPLORE) algorithm [26] induces decision rules systematically in disjunctive normal form (DNF). It employs a branch-and-bound strategy that allows for user-defined performance constraints. By employing exhaustive search to learn rules in DNF form, EXPLORE offers an alternative to greedy search. It produces noticeably smaller classifiers while maintaining comparable or occasionally even superior performance. The exhaustive search of the feature space approach utilised by EXPLORE allows for the examination of all possible classifiers, and the best classifier with least complexity and the best optimization value under the user defined constraints is selected. However, the drawback with exhaustive search algorithms is that the execution time quickly becomes prohibitive even for moderately sized problems and EXPLORE is no exception.

The majority of rule learners favour accuracy over other metrics, however this is not always the optimal approach. Sensitivity is particularly crucial in specific application areas, such as the healthcare industry. In medicine, the goal is to prevent members of the positive class from slipping away, and sensitivity becomes the preferred metric because it shows how much of the positive class the model is able to capture [55]. EXPLORE offers versatility, allowing users to choose their own performance metrics and to specify the minimum constraints that a rule must achieve. EXPLORE can be optimised on any one of the following performance metrics, sensitivity, specificity, positive predictive value, negative predictive value or accuracy. Also, minimal constraints on one or more of the performance measures that should be attained by the rule can be set. To induce a DNF rule from examples with EXPLORE; a performance measure, minimum constraints and the rule length are specified. EXPLORE systematically generates all rules of the specified length. The best rule is selected, i.e., the rule for which performance metric is maximal and minimum constraints are met.

EXPLORE offers direct control of the rule complexity by allowing a maximum rule length to be set. As discussed in section 3.4 the rule complexity is the total number of literals in the DNF rule and maximum rule length allows the user to limit it. The resulting classifiers of most rule-based models are often complex. It is possible for smaller and more comprehensible classifiers to achieve comparable performance to the complex ones. When dealing with a sensitive and vulnerable population, comprehension is critical since the reasoning behind the model's judgements is crucial to the users. Poor comprehensibility is related with ordered rule sets since all preceding rules to a particular rule contribute to the interpretation of that rule, making a single rule in and of itself insufficient. A single rule written in Disjunctive Normal Form (DNF) would be lot easier to grasp, and that is exactly what EXPLORE does. EXPLORE generates a single rule of desired complexity in DNF format that is easily understood by non-technical users while performing comparably to more complex models.

```
EXPLORE Algorithm
=========================================================================================
Input : X,F,n,P,C
Output: b
=========================================================================================
1  FO = GenerateFeatureOperatorList(F );
2  V  = GenerateValueLists(X, FO);
3  T  = (n);
4  L  = null;
5  b  = null;
6  do
7      InitFeatureOperators(FO, L, T );
8      do
9          if InitValues(L, T , V ) then
10             do
11                 if ( P(r, X) > P(b, X)) & C(r, X) then
12                     b = r ;
13                 end
14             while NextValues(L, T , V )
15          end
16     while NextFeatureOperators(FO, L, T )
17 while NextTermTuple(T )
18 return b;
```

Lets briefly describe the DNF rule induction process of EXPLORE. The algorithm takes as inputs, X (the examples), F (feature names), n (rule length), P (performance measure) and C (performance constraints) and generates b (the best rule) after an exhaustive search. This process can be summarised as follows;

1. **Initialization (lines 1 - 5)**: An initialization process precedes the actual rule search. Feature - operator pair lists (FO), are generated and sorted alphabetically by the generate feature operator list

function in line 1. The input to this function are the list of feature names (F). For each feature, if it is categorical, the tuple (f, =) is appended to the list otherwise (f, >) and (f, ⩽) are appended to the list. From Table 3.1 the list will be;

$$\text{FO} = \begin{bmatrix} (\text{Humidity}, >) \\ (\text{Humidity}, \leqslant) \\ (\text{Weather}, =) \\ (\text{Wind}, =) \end{bmatrix}$$

In line 2 a list of values for instantiating the literals are then generated by the function, generate value lists which takes the examples and the feature operator pair list as inputs. For each "FO" if the feature is categorical then all its unique values are appended. For continuous feature, all the averages of two subsequent values in a sorted list are taken. A technique called subsumption pruning is employed to reduce the number of mid-point values considerably without loss of performance. The idea of subsumption pruning is that a literal is removed from the set of promising literals if there exists another literal that covers a superset of the positive examples and a subset of the negative examples [40]. The values then sorted in decreasing order by the number of true positives generated by each single literal (thus feature, operator, value triad). From Table 3.1 the value list will be,

$$\text{V} = \left\{ \begin{array}{lcl} (\text{Humidity}, >) & : & [27, 43, 46.5, 47] \\ (\text{Humidity}, \leqslant) & : & [47, 45, 36] \\ (\text{Weather}, =) & : & [\text{Sunny, Cloudy}] \\ (\text{Wind}, =) & : & [\text{Strong, Weak}] \end{array} \right\}$$

In line 3, term tuple (T) is initialised as a tuple with the rule length (n). The term tuple describes the logical combinations of the literals in the rule. It is a list of the term sizes of the rule. For instance, T=(2,1) implies that the rule is a disjunction of 2 terms, the 1$^{\text{st}}$ term is a conjuction of 2 literals and the 2$^{\text{nd}}$ term is a single literal. The generation of term tuples is explained in step 7.

In line 4 and 5, literal (L) and best rule (b) are initialised as null. Literal (L) is the list of literals that will be used to instantiate the rule. Best rule is a container that stores the current best performing rule. Any time a new rule out performs the current best rule then it is replaced by the new rule in the container.

2. **Feature - operator instantiation (lines 6, 7)**: The actual rule search begins here. The function

init feature operators, initialises the feature-operators in the literal tuples. For each term, if it is the first term or the term's size is not equal to the preceding term's size, then the term is initialised to lexicographically ordered feature-operators. If the term sizes are the same and greater than one, the term is initialised by replicating the previous term's instantiation. If term sizes are equal to one, the preceding term's instantiation for a categorical feature is replicated; for a continuous-valued feature, the term is instantiated with next feature-operator.

For illustration, let n = 3; the starting term tuple (T) will be T=(3). The init feature operators function will return;

$$\mathrm{L} = \begin{bmatrix} (\text{Humidity}, & >) \\ (\text{Humidity}, & \leqslant) \\ (\text{Weather}, & =) \end{bmatrix}$$

3. **Value initialisation (lines 8, 9)**: Here the literals, which are only feature-operators, are now instantiated with values. The number of true positives of a literal comprising the feature-operator and the value was used to sort the values of each feature-operator in decreasing order. The first value in the value list is used as the initial value for all literals because the literal with the most true positives has the greatest potential to improve the rule. The init values function will return;

$$\mathrm{L} = \begin{bmatrix} (\text{Humidity}, & >, & 27) \\ (\text{Humidity}, & \leqslant, & 47) \\ (\text{Weather}, & =, & \text{Sunny}) \end{bmatrix}$$

4. **Rule evaluation (lines 10 - 13)**: At this point the rule is fully instantiated with complete literals and terms and can be evaluated. The fully instantiated rule will be;

$$\mathrm{r} = \left(\begin{pmatrix} (\text{Humidity}, & >, & 27) \\ (\text{Humidity}, & \leqslant, & 47) \\ (\text{Weather}, & =, & \text{Sunny}) \end{pmatrix}, \quad (\ 3\ ) \right)$$

This rule can be interprated as the conjunction of all three literals. If the current rule out performs the stored best rule and meets the conditions of the constraints then the stored best rule is replaced with the current rule and the next set of values are instantiated (step 5) for evaluation again. If the performance of the current rule is less than the stored best rule or could not meet the conditions of the constraints then the next set of values are instantiated (step 5) for evaluation again. At this stage since b = null, automatically this rule becomes the best rule.

5. **Next set of values instantiation (line 14)**: Only values that have the potential to achieve a performance greater than the current best performing rule are routinely used to instantiate the literals in the rule. A branch-and-bound strategy is used to accomplish this. The performance metric that needs to be optimised and the user-defined performance constraints both influence the bounding rules. The function next values, instantiates the rule with the next set of values that fulfill the bounds. The bounds of EXPLORE are much more elaborate, but for simplicity lets define our bounds as FP$\leqslant$3. Starting with the last term, it instantiates the literal tuples using the next set of values for that term. In the event that this is not feasible, the search proceeds up, until a term that can be instantiated with the next set of values is discovered. For a term that can be instantiated, all terms below it are initialised again . The possibility exists, though, that one of these terms cannot be initialised to a value that satisfies the bounds. In that instance, next values for the term are determined and we try again. If a rule that meets the bounds is found, the function returns True; if not, it returns False. The rule after the next set of values have been instantiated will be;

$$\mathrm{r} = \left(\begin{pmatrix} (\text{Humidity}, & >, & 27) \\ (\text{Humidity}, & \leqslant, & 47) \\ (\text{Weather}, & =, & \text{Cloudy}) \end{pmatrix}, \quad (\ 3\ ) \right)$$

Step 4 and 5 are repeated until the value space is exhausted.

6. **Next set of feature-operators instantiation (line 16)**: When the value space has been exhausted with the current rule, the function next feature operators instantiates, the rule with the next set of feature-operators. Moving up from the last term, the first term for which a next set of feature-operators can be generated is identified. This term is instantiated with the next set of feature-operators, and all terms below it are initialised. The instantiation of the identified term is done literal by literal. For this term, the first literal that is not instantiated with its last possible feature - operator is found by starting at the last literal in the term and proceeding upwards. The next feature-operator is used to instantiate this literal, and all feature-operators below it are initialised. The rule after the next feature operators has been instantiated will be;

$$\mathrm{r} = \left(\begin{pmatrix} (\text{Humidity}, & >) \\ (\text{Humidity}, & \leqslant) \\ (\text{Wind}, & =) \end{pmatrix}, \quad (\ 3\ ) \right)$$

The rule is then initialised with values in step 3. Step 3 through step 6 are repeated until all the literals have been instantiated with their last possible feature-operator then the next term tuple is generated in step 7.

7. **Next term tuple generation (line 17)**: This is simply a positive integer partitioning which represents, how the literals should be combined logically in the rule. The first term tuple was initialized to n = 3 (the rule length). This function generates the next the next term tuples systematically. The first time this function is called it generates T=(2,1), on the second call T= (1,1,1) and then exits on the third call since it has reached the lowest integer partition of 3. Below are all the term tuples.

```
(3)
(2,1)
(1,1,1)
```

   Once a next term tuple is generated successfully, all the literal tuples are then initialised with feature operators in step 2. The cycle from step 2 to step 7 are repeated until the last term tuple has been generated. Finally the algorithm returns the best performing rule.

4.4 Summary

This chapter offered a quick overview of the algorithms under consideration for this study. CART, a fundamental model for interpretability, does a simple threshold test on all the midpoint values of a continuous feature and splits on the value with the highest homogeneity. A novel method, GOSDT, builds optimal decision trees for a variety of objectives. GOSDT splits at every mid-point value for continuous values features, but to decrease the search space, ”similar support” bounds are employed. EXPLORE is an exhaustive technique that generates decision rules systematically in disjunctive normal form (DNF) using a branch-and-bound strategy that allows for user-defined performance measures and constraints. Furthermore, EXPLORE’s maximum rule length enables for direct control of the rule complexity. Finally, the EXPLORE algorithm and the rule induction process is explained in Section 4.3.

Chapter 5

Results and Discussion

As mentioned previously, choosing the “best model” for the given classification task depends not only on discriminatory power, but also on other factors such as ease of implementation, model’s complexity and complexity control. These factors are intertwined and very necessary for this work. As discussed in section 1.5, classification rules in DNF format combines all these attributes inherently, hence the models under study either generates rules in DNF form or can be translated to rules in DNF form. In this chapter, the research results are reported and interpreted in the context of this work and why they matter. The remainder of the chapter is organised as follows; Section 5.1 provides a description of current efforts to connecting chronic homeless individuals with housing support and the proposal of this work to augment the current efforts, Section 5.2 describes important factors considered in generating the results presented, the effectiveness of controlling the complexity of the classification DNF rules of the tree models is discussed in Section 5.3, Section 5.4 discusses statistical metric evaluation of the performance of the models reviewed, in Section 5.5 a real time program delivery approach experimental results are discussed, how well the models generalise is examine in Section 5.6 and finally the the best performing rules in DNF format are presented in Section 5.7.

5.1 Solution Application in a Shelter System

There are several initiatives aimed at identifying and connecting chronic homeless people with housing resources. Street outreach programmes are one such way. Street outreach, defined as contacting people on the street to increase their access to resources, is a key means of directly engaging homeless people and providing them with the housing and health care services they require. Street outreach has long been an important practical approach for discovering homeless persons and connecting them to social and housing programs. Several studies [56][57] have demonstrated the efficacy of outreach in connecting homeless populations to services, including housing and health care.

Another method for referring homeless people for housing assistance is; coordinated access and assessment (CAA)[58]. Coordinated Access and Assessment is the system that connects homeless persons with the housing and services they require. Such a programme is run by the Calgary Homeless Foundation (CHF). Housing strategists are caseworkers that are trained to connect individuals and families to information, resources, and housing. A homeless person's level of need is assessed using a set of standardised instruments known as the needs and services questionnaire and the housing plan.

An alternative approach is brought forth to augment these other efforts in identifying homeless individuals and doing it quickly; for permanent housing support. Using artificial intelligence to aid social workers in the fight against homelessness is a revolutionary way to go. Emergency shelter staff do not have data science expertise. Therefore to achieve this goal a workflow that creates collaboration between the emergency shelter staff on one hand and data scientists on the other hand to harness their unique expertise for the betterment of the homeless population is devised.

Data scientists will use the recommended model to generate the decision rules in disjunctive normal form, limiting complexity as specified by shelter staff. These decision rules will then be made available to the emergency shelter staff to implement by hand on a simple spreadsheet or any other database system at their disposal. As discussed in Section 1.3, it is important that the solution encompasses real time program delivery. Therefore the decision rules will be applied on a continuous basis at regular intervals . Let the decision rules application interval be denoted by T_L. An individual that has interacted with the shelter for the 1^{st} T_L days, when the decision rule is applied on the person's data, it should be able to effectively predict whether the person is at risk of becoming a long-term shelter user and referred to human counsellors for consultation and final decision making. If the person is not identified to be at risk; after another T_L days, if the same person is still interacting with the shelter, the same decision rule should still be able to classify the person by applying it on the person's $2T_L$ data. If the person is not identified to be at risk; after further T_L days, if the same person is still interacting with the shelter, the same decision rule should still be able to classify the person by applying it on the person's $3T_L$ data. This process will be repeated after every T_L days for any client that is still interacting with the shelter until the person exits the shelter or is identified to be at risk of long-term shelter use or the efficacy of the rule ends. The shelter staff should also know when the efficacy of the rule ends and to use new decision rule on persons still interacting with the center after the old decision rule failed to identify them on successive attempts.

5.2 Results Generation

Several factors were taken into account in producing the results in this section for discussion. These factors are instrumental to generating the best results for discussion.

5.2.1 Overfitting

Overfitting is a major issue while training machine learning models. Overfitting arises when a model performs exceptionally well on the data used to train it but fails to generalize successfully to new, previously unseen data points. This could be due to noise in the data or because the model learned to predict certain inputs rather than the predictive parameters that would help it make good predictions. Disjoint data samples were created for training and testing the models to avoid overfitting. One set is known as the train set, and the other as the test set. The train set was then used to train the models, whereas the test set was a collection of data points on which the models were evaluated. To offer the models with as much data as possible to discover meaningful patterns from, the training data was far greater than the testing data. The split was done such that the train set comprised of 70% while the test set 30%.

5.2.2 Stratification

The proportion of the class distribution was taken into account when separating the data into train and test sets, allowing the algorithms to learn from a data set that is a true reflection of the primary data set. Stratified sampling was employed specifically, to generate train set and test set with populations that best represent the true distribution of the whole population under investigation. Stratified sampling is a method of selecting samples in the same proportion as they appear in the population. To ensure that the train set and test set are representative of the overall population, stratified sampling divides the population into subgroups or strata and samples the required number of examples from each stratum. The stratification was done over the labels since the long-term shelter users are of particular importance to this investigation. It is therefore necessary that the stratification was done over the labels when splitting the data into train set and test sets. It is important to mention that stratifying over the labels ensures that the proportion of long term clients in the primary dataset remains the same in the train set and the test set but the splitting of the data is done by examples (clients) and not by data features.

5.2.3 Fairness

To be fair to the competing machine learning algorithms, they must be trained and assessed on the same subsets of the dataset. This was taken into account by using the same train set to train all of the models and the same test set to evaluate all of the models. For this work predictive accuracy is not the only measure we are aiming for. The complexity of the models and the degree of control provided by the algorithms to generate decision rules of varied complexities were both studied. The depth of the decision trees were limited to a maximum height of two, three, and four while training the tree models. Because a tree height of one is simply the root node, and only the root is not a tree and provides insufficient information, it was not investigated in the study. As mentioned in Section 1.4, the human cognitive memory can process 7 ± 2 items at a time, it will be observed later in Section 5.3 that even a tree height of four when translated to DNF format can be quite complicated, resulting in the decision to limit the study to no more than a tree height of four. In training EXPLORE, it was allowed to generate decision rules of length one to five, where rule length is the number of literals contained in the decision rule.

5.2.4 Tree Pruning

Pruning the tree models is critical if the decision trees are to compete with EXPLORE on all fronts (performance and complexity). Large decision trees are more prone to overfitting, and proper pruning can lessen this likelihood, improving performance. A smaller decision tree will also result in a smaller DNF rule. Pruning decreases the size of decision trees by deleting branches that lack the ability to classify instances. Both pre-pruning and post pruning were employed to control the complexity of CART and GOSDT. Section 3.5 describes the post pruning method used in this work. The pre-pruning method employed to control the CART models' complexity was by limiting the height of the tree. This method stops the decision tree from growing more complex once the set height has reached. A combination of two pre-pruning methods was employed to control complexity of the GOSDT models; setting a regularization parameter that penalizes more complex models and also setting a maximum height that allows the model to generate shorter trees with respect to the set height. A bigger regularization value penalizes the model the most leading to a sparser decision tree.

5.2.5 Data Normalization

To train a machine learning model on one observation window size dataset and use that model to make predictions effectively on a different observation window size dataset, the features of both datasets needs to be standardized to a unified scale without lose of information. For instance the 90days window size data; contains client records for 90 days while the 60days window size data; contains client records for 60 days. Hence the features of different observation window size datasets will have different scales. A technique called normalization was employed to standardize all the datasets used to generate the results in Section 5.5.

The goal of the normalization is to change the values in the dataset to a common scale, without distorting differences in the ranges of values or losing information. Normalization creates new values that maintain the general distribution and ratios in the source data, while keeping values within a scale applied across all columns. Since each observation window size dataset holds records for a duration equal to the window size, a simple normalisation was done as follows; ($\frac{\text{Feature}}{\text{Window Size}} \times 360$). Multiplying by 360 ensures that the normalised feature values still remain as non-negative integer values. For example a normalised version of Table 2.4 is shown in Table 5.1

Client ID	Event__Sleep	Event__Mail	Addiction
001	24	12	60
002	12	12	36
003	12	0	0

Table 5.1: Illustrative normalised 30days window size data

5.3 Rule Complexity and Complexity Control

Figures 5.1 and 5.2 are representations of the various heights of trained CART and GOSDT decision trees and the corresponding complexities of their DNF versions after translation. In Section 3.4, decision rule complexity is defined as the total number of literals in the rule. Conversely decision tree height is defined as the length (number of nodes or literals) of the longest path from the tree root to a leaf. CART and GOSDT were trained on all the different cohorts' 60 days window size data. For each cohort the algorithms (CART and GOSDT) were trained for three different maximum tree heights (thus; 2, 3 and 4). The generated models were then translated into DNF format following the technique discussed in Section 3.5 and the various rule complexities calculated. As discussed in Section 3.4, minimising DNF rule complexity have been described as NP-Complete and most works have focused on determining the smallest DNF formula. It is however worth noting that this work did not explore minimising the DNF rules for all algorithms to a minimum number of literals or terms.

It is evident that the complexities of CART and GOSDT DNF rules increased rapidly as the height of the decision tree increased. Though the pruning methods employed have drastically reduced the complexity of the CART and GOSDT models, clearly in these figures it can be seen that these options are however not very efficient when the model is translated to an easily implementable form; DNF format. EXPLORE grants direct access to control the complexity of the rule generated in DNF form. EXPLORE allows for the desired number of literals (rule length) in DNF form that the generated rule should contain, to be set. All decision rules of the specified length are generated, and the best performing decision rule selected.

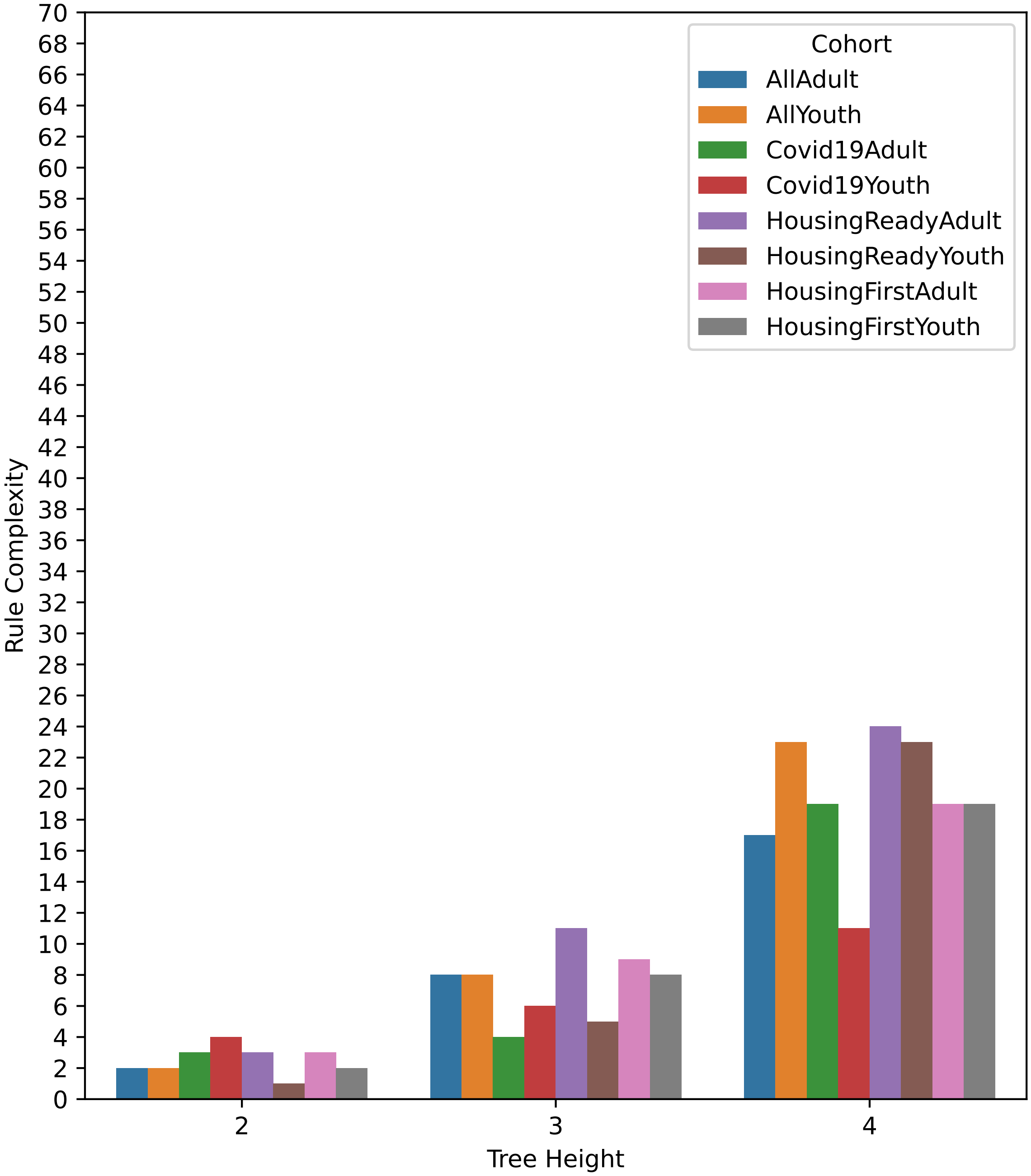

Figure 5.1: CART tree height Vs rule complexity

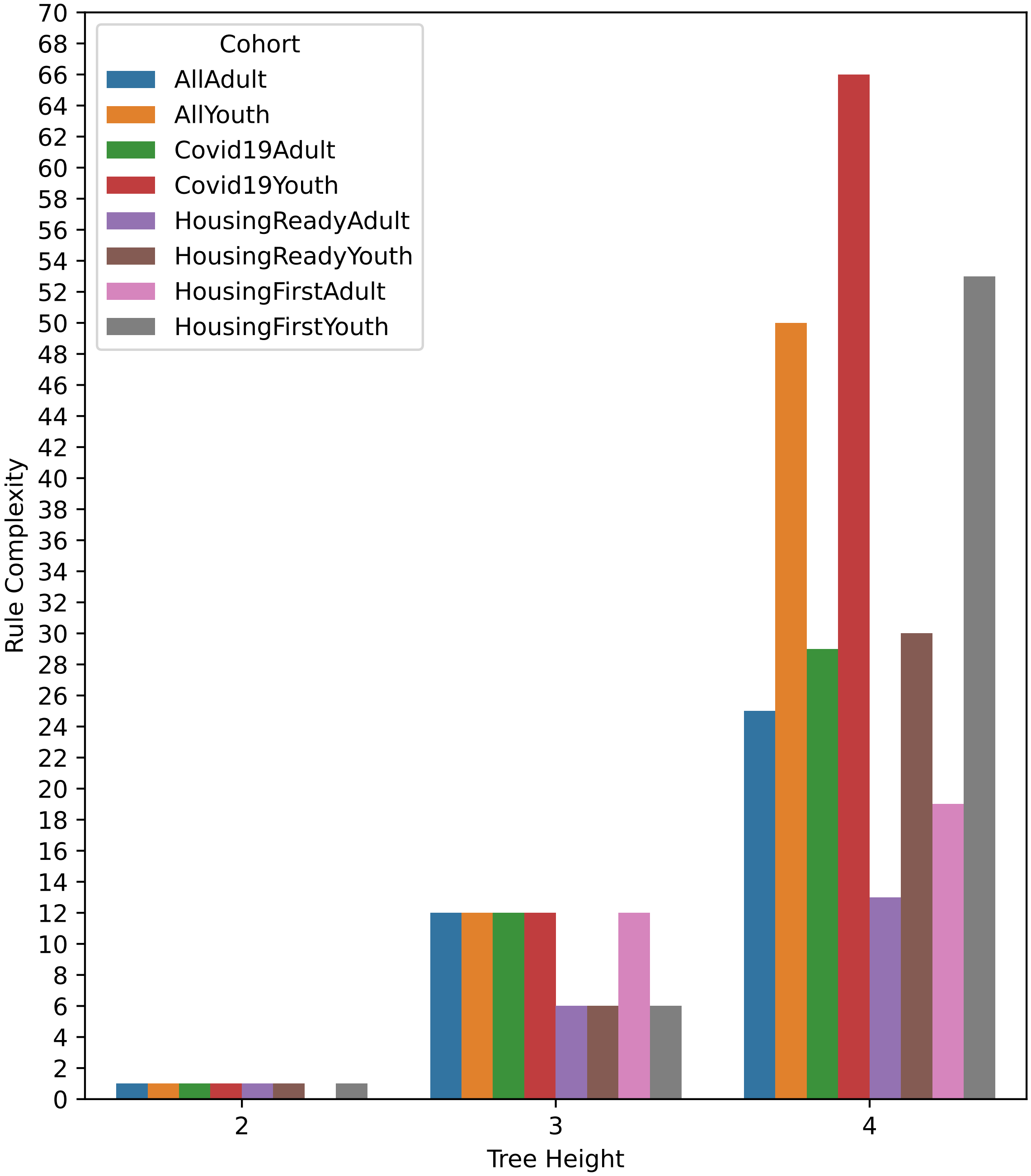

Figure 5.2: GOSDT tree height Vs rule complexity

It can also be seen from the figures that the different models generated by CART and GOSDT when trained on different datasets for the same tree height are of different complexities. Dataset characteristics has impact on the complexity of logical models. These data characteristics could be the type of data features in the dataset, the number of features in the dataset or the number of unique values each feature in the dataset has. It can be seen from the figures, how the data characteristics impacts on the complexities of the models generated by CART and GOSDT. Hence these models (CART and GOSDT) have no direct control of the complexity of the output models as far as the data factor is concern. As mentioned, the desired decision rules should be less complex and very discriminatory as well. Scaling down the data to help produce less complex decision rules will sacrifice optimality and may also introduce bias, hence not a viable alternative to controlling complexity for this work. The rule length parameter of EXPLORE is a robust and hard condition and not affected by data size or data characteristics, hence once set to a desired value there is guarantee that the output decision rule in DNF format will be of that exact complexity.

5.4 Metric Evaluation

Different metrics; thus precision, recall and F_1Score were employed to understand how the models performed. Let us first introduce the prediction outcomes. There are four possible outcomes based on the truth values of the actual and predicted classes. For a positive example, if the prediction is also positive; it is a true positive, if the prediction is negative; it is a false negative. For a negative example, if the prediction is also negative; it is a true negative, and a false positive if the prediction is positive. Precision $(\frac{\text{True Positives}}{\text{True Positives + False Positives}})$ tells us the fraction of positive predictions that actually belong to the positive class. This metric is often used to determine how confident a model is at its predictions. Precision is a valuable indicator since it shows the fraction of referrals who would become long term shelter users, since (True Positives + False Positives) represents the number of housing suggestions. Recall $(\frac{\text{True Positives}}{\text{True Positives+False Negatives}})$ tells us the fraction of actual positives that were identified correctly. This metric is often used to determine how sensitive the model is to the positive class. High recall is essential because missing anyone who requires support is not desirable. The F_1Score corresponds to the harmonic mean of precision and recall. A comprehensive discussion of these metrics in presented in Appendix A

5.4.1 Performance with Complexity Constrains on Algorithms

All Adult Cohort

A trade-off between model complexity and model performance was investigated. Ideally a less complex model with comparable performance against a more complex model is desirable. Plots of the performances of models of varied complexity levels trained on all adult window size 90 days cohort data are represented by Figures 5.3, 5.4 and 5.5. As described in Section 5.2.2, the all adult window size 90days cohort data was split into 70% train set and 30% test set. The train set was used to train all the algorithms and the test set to evaluate all the generated models. CART and GOSDT were trained by using tree heights of 2, 3 and 4. Also a regularization of 0.1 was set on GOSDT to assist it build even more sparser models. EXPLORE was trained by using rule lengths of 1, 2, 3, 4 and 5. The generated models were then evaluated on the test set. Furthermore, the generated models of CART and GOSDT were translated to DNF rules using the method described in Section 3.5 and the various rule complexities determined. The complexities of the various rules were plotted against their respective performance values.

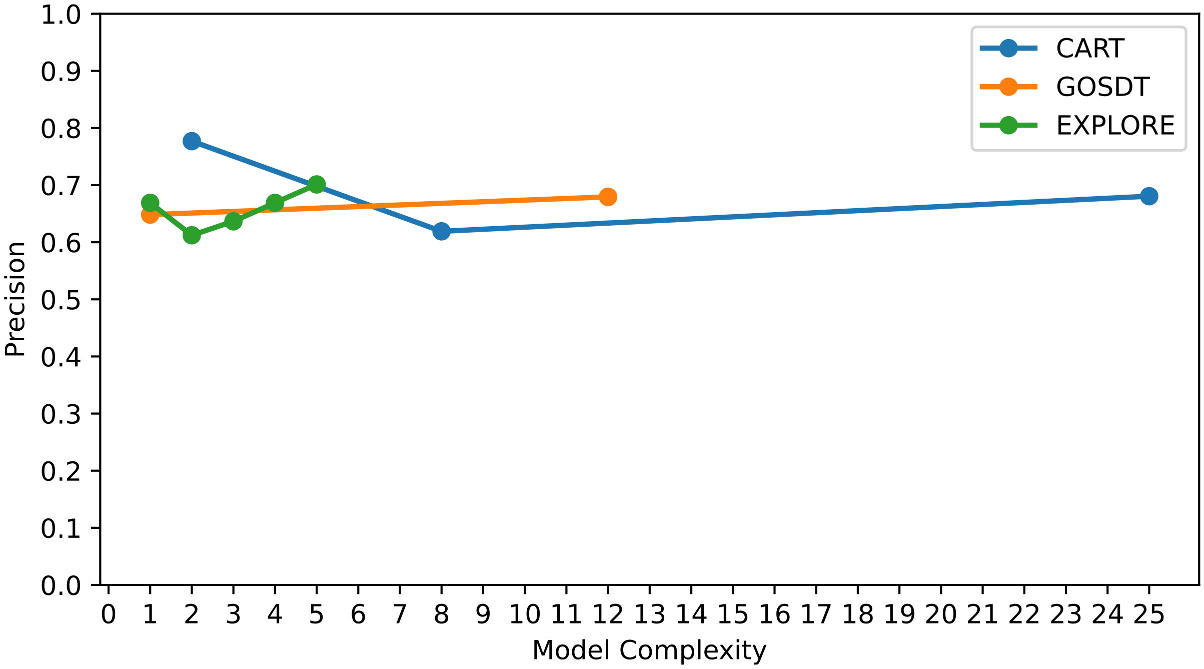

Figure 5.3: Precision with increasing complexity when trained on the all adult window size 90days cohort

The findings in Figures 5.3 - 5.5 indicate that machine learning alone is not enough to alleviate homelessness. The recall plot in Figure 5.4 highlights that the algorithms still overlooked roughly 50% of long-term shelter users on average. As discussed in Section 5.1, various other programmes exist to identify chronic

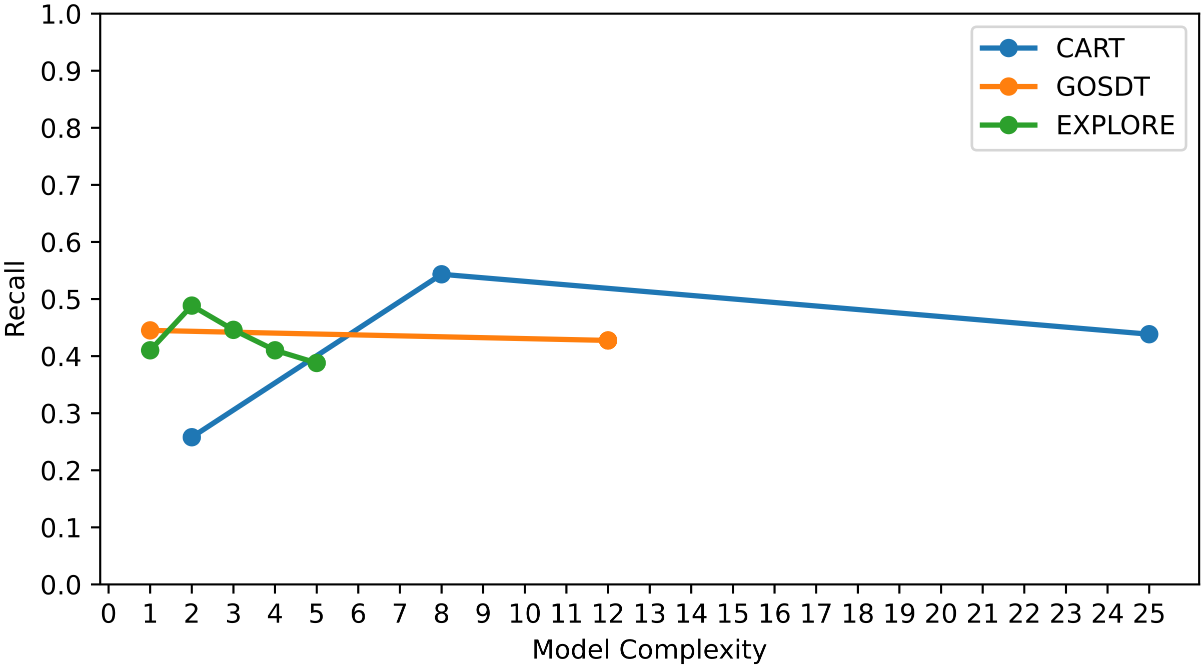

Figure 5.4: Recall with increasing complexity when trained on the all adult window size 90days cohort

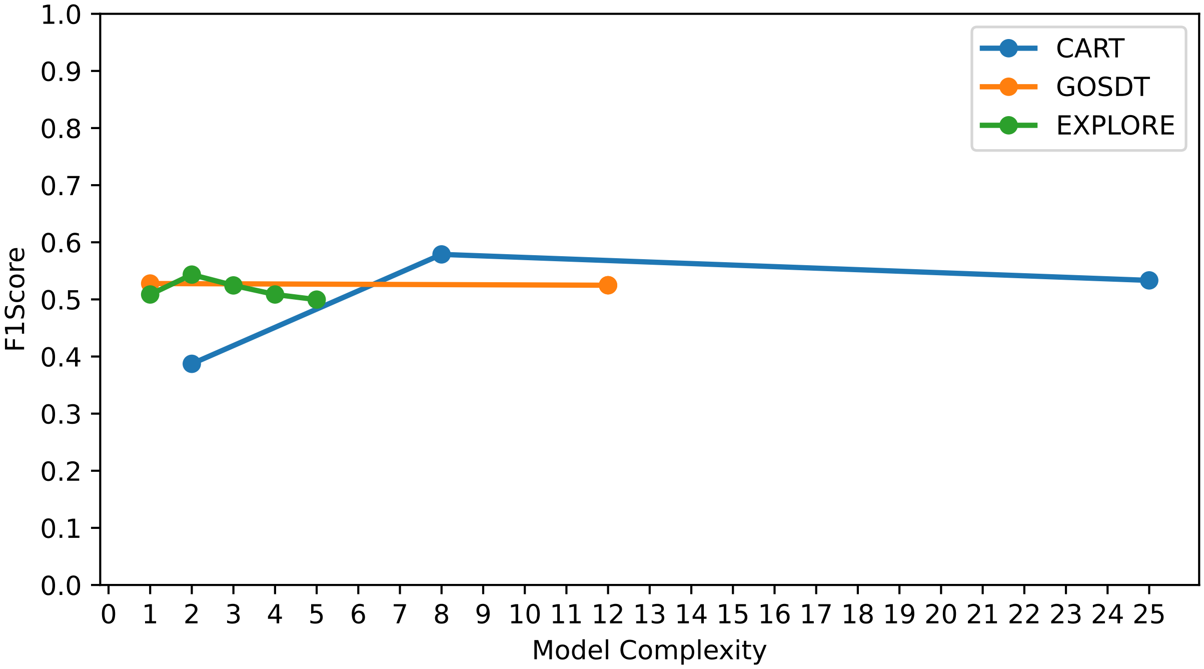

Figure 5.5: F_1Score with increasing complexity when trained on the all adult window size 90days cohort

homelessness for housing assistance, and the purpose of our study is to supplement current efforts. Though 50% is a substantial miss, the individuals identified to be at risk of long term shelter use, by the machine are done really quick. One of the most significant advantages of machine learning in the fight against chronic homelessness is the early identification of long-term shelter users. The various programs rely on the traditional definitions outlined in Section 2.3.2 to identify chronically homeless persons. By these traditional definitions, for an individual to be considered chronically homeless, the person must be homeless for a minimum of 180 days in the past year. These decision rules can identify a person within 90 days into the person's homelessness. Secondly, a subset of the long-term shelter users which I refer to as 'silent clients' stand to gain the most from machine learning solutions. These 'silent clients', accumulates a large number of shelter stays without drawing attention to themselves. They are not at the forefront of shelter staffs' minds due to their few encounters with shelter staff and may otherwise slip away unnoticed.

Though Figure 5.4, shows that the CART model of complexity level 9, refers a higher proportion of long-term shelter users for housing assistance but it is actually casting a wider net, picking a greater number of true positives but also a greater number of false positives. Secondly, as mentioned in Section 1.4, humans are only capable of managing 7 ± 2 cognitive entities at once. Having said that, a rule complexity of 9 can be considered interpretable but it is till very hard on humans. EXPLORE which offers full control of its rule complexities is still closely competing with its smaller models.

Precision and recall however cannot be utilised separately for resource allocation because they are interdependent. It is also worth mentioning that the generated rules by GOSDT for tree heights of 3 and 4 are the same. Hence, it can be seen in the figures that GOSDT has only two data points. Furthermore, Figure 5.3 shows that the algorithms are quite efficient in directing resources to those who need them the most.

All Youth Cohort

Plots of the performances of models of varied complexity levels trained on all youth window size 90 days cohort data are shown in Figures 5.6 - 5.8. As described earlier in this Section the same data splitting, training and testing procedure for generating the all adult window size 90 days cohort results was applied to the all youth window size 90 days cohort data to generate the results in the Figures.

The ultimate goal of this work is to prevent as many youth as possible from being connected to the streets. Similar to the all adult results presented earlier in this Section, the all youth rules are capturing on average 40% of the youth at risk of long term shelter users. This is clearly not enough but as discussed in Section 5.1 the machine learning solution is a supplement to existing methods. While other methods might take a considerable amount of time and effort to identify the youth at risk of long term shelter use, the machine does it in a timely fashion. The subset of youth suggested by the machine can benefit from support

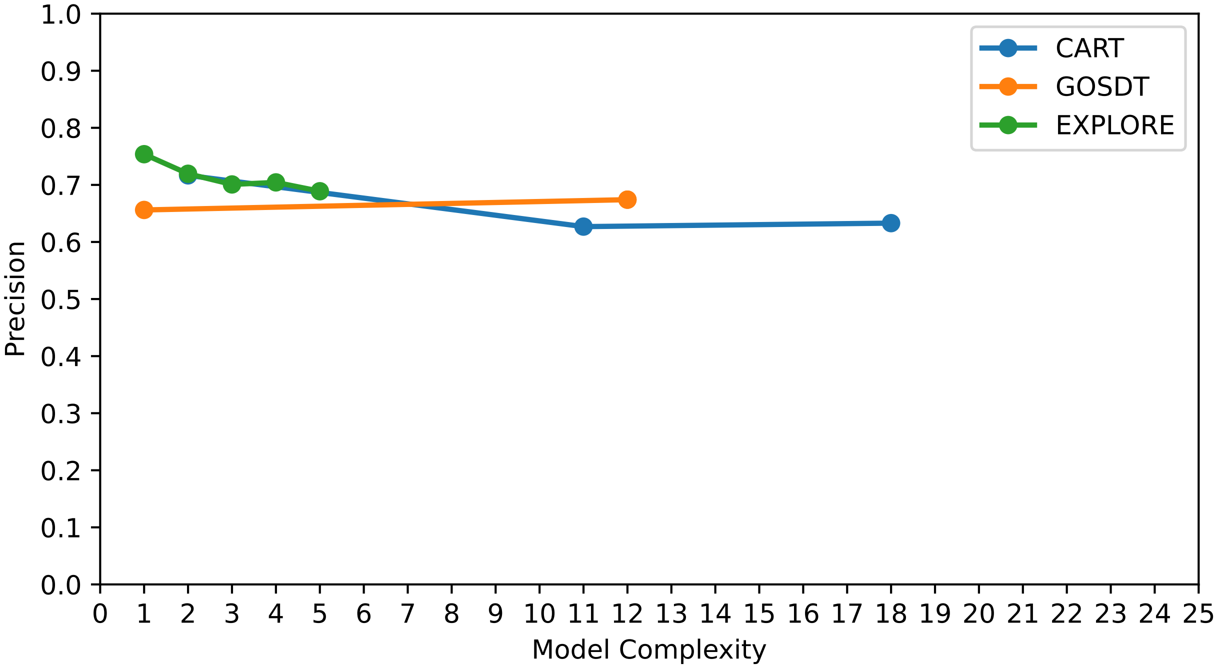

Figure 5.6: Precision with increasing complexity when trained on the all youth window size 90days cohort

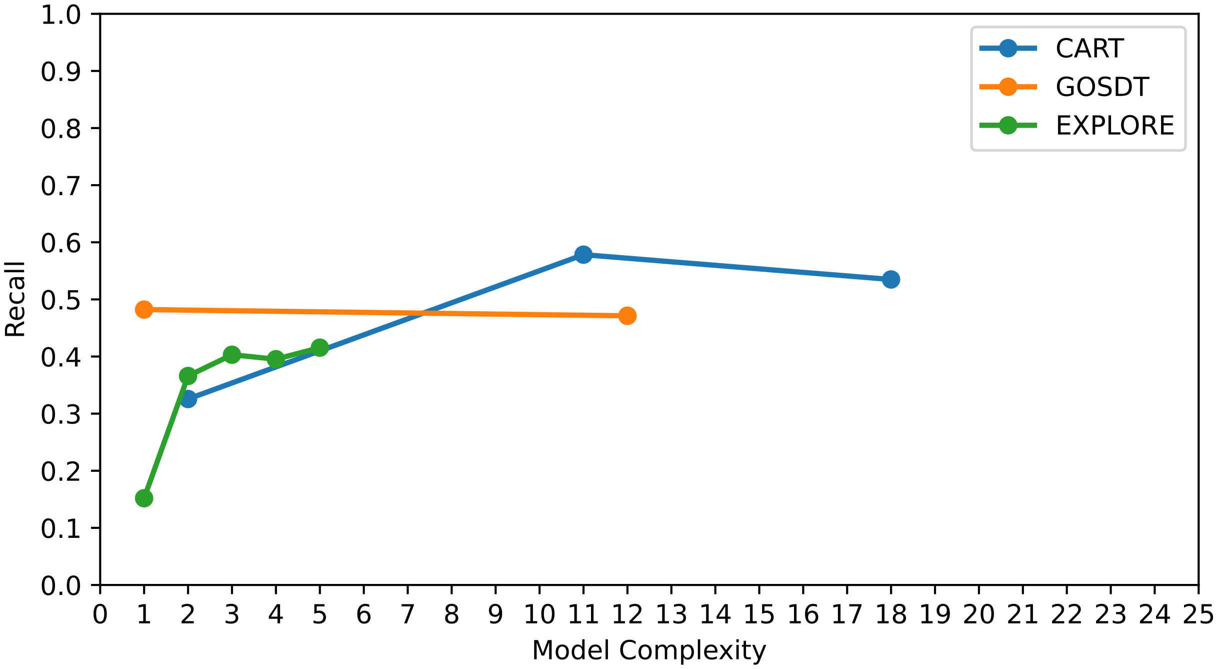

Figure 5.7: Recall with increasing complexity when trained on the all youth window size 90days cohort

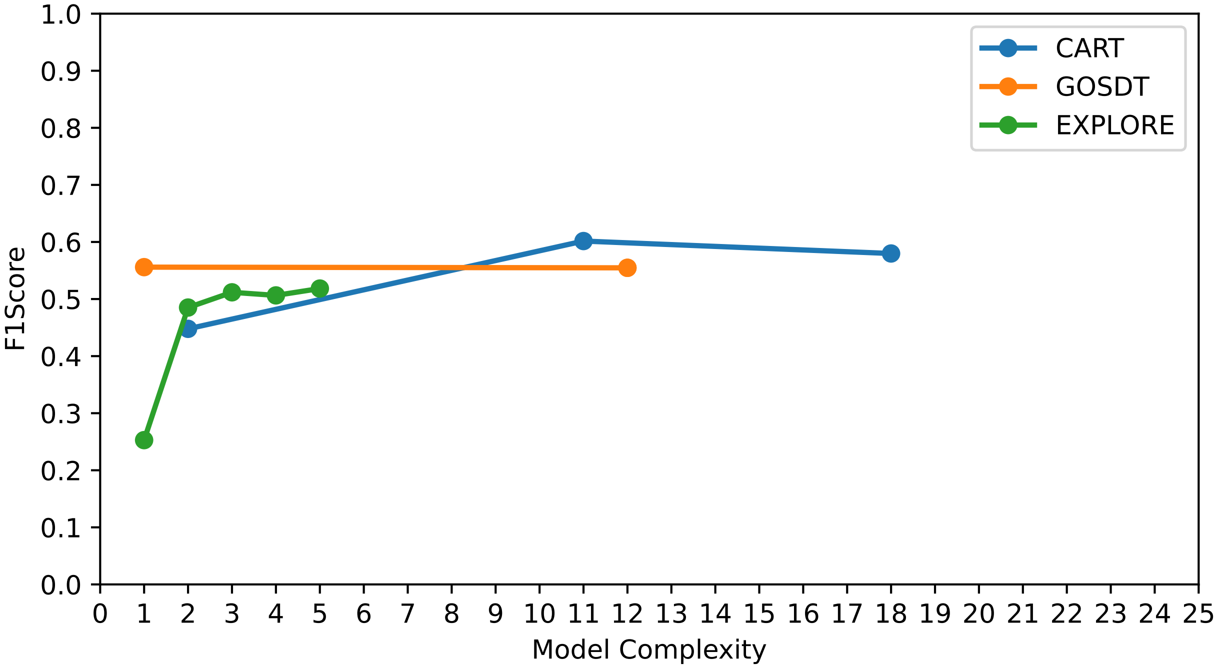

Figure 5.8: F_1Score with increasing complexity when trained on the all youth window size 90days cohort

before their mental and physical conditions deteriorate.

Interestingly extra complexity is not improving the performance of the algorithms. The reason could be that, only a few of the data features are highly discriminatory. Implying that most of the youth interacting with the DI center are only accessing a minority of services provided by the center.

Also for the same tree heights, the tree models are generating rules of different complexities when trained on the all youth window size 90 days cohort compared to when trained on the all adult window size 90 days cohort. As stated in [59] and [60], dataset characteristics is one of the major impact factors on the complexity of logical models. These data characteristics could be the type of data features in the dataset, the number of features in the dataset or the number of unique values each feature in the dataset has. In this instance, the all youth window size 90 days cohort and the all adult window size 90 days cohort have different number of unique feature values which could be the cause in model complexity disparity.

5.4.2 Performance without Complexity Constrains on Algorithms

The performance of the algorithms with no constrains on complexity is shown in Table 5.2. To generate these results the 90 days all adult cohort data was split into disjoint sets as described in Section 5.2. The algorithms were trained on the same train set and applied on the the same test set. No complexity restriction was imposed on the algorithms. The trees of CART and GOSDT were allowed to keep splitting and growing until no further split improved performance. The trees translated into DNF format as described in Section 3.5 and the complexity determined. EXPLORE was allowed to train with incremental rule lengths, comparing the performance measure of the best rule of each rule length to the best rule of the previous rule length. The performance measure set out for EXPLORE was recall. If the performance measure of the current rule length's best rule is greater than the previous rule length's best rule the rule length is incremented by 1, otherwise the previous rule length and performance is returned.

Though logical models can lose their interpretability with large number of literals, it was worth investigating whether interpretability could be traded for higher discriminatory power. The results in Table 5.2 shows that less complex models still have comparable predictive power in addition to being interpretable. For instance CART is still missing about 60% of the long-term clients despite the fact that it generated a very large tree which is impossible to comprehend by any human, let alone implementing it by a non-technical person.

Algorithm	Rule Complexity	Precision	Recall	F1 Score
CART	19042	0.6132	0.4166	0.4961
GOSDT	1	0.6210	0.4912	0.5485
EXPLORE	1	0.6767	0.3747	0.4823

Table 5.2: Performance of models without complexity constrains

5.5 Real Time Program Delivery

Figures 5.9 to 5.11 shows the performance of window size 60days' rules applied on all window sizes of the all youth cohort. To generate these results the 60, 90,120 and 180 days all youth cohort data were normalised, following the description provided in Section 5.2.5. The normalised all youth window size 60 days data was split into 70% train set and 30% test set following the description provided in Section 5.2.2. All the clients in the all youth window size 60 days train set were removed from the normalised all youth window size 90,120 and 180 days data sets. The purpose of this is to prevent the models from predicting clients they have seen already during training. This can cause the models to overfit, by just memorizing the positive clients in the train set. Secondly, making predictions on the same clients across all models offers a fair play ground and is the right way to apply machine learning. The remaining normalised all youth window size 90,120 and 180 days data sets containing the same clients as in the all youth window size 60 days test set were used as additional test sets.

All the algorithms were trained on the all youth window size 60 days train set to generate classification rules. These decision rules were then used to make predictions on the all youth window size 60 days test set. The performance of the models were recorded. All clients that the models identified to be at risk of long-term shelter use were removed from the subsequent window size test sets (normalised 90, 120, and 180 days test sets). The reason being that; it is assumed that all identified clients at risk of long-term shelter use are referred for housing support and they ceased to interact with the DI center.

The same 60days decision rules are then used to make predictions on the remaining normalised window size 90days test set. The Idea is that a long-term shelter client that was not identified after 60 days of interaction with the shelter could still be identified after 90 days of interaction with the shelter by the same decision rule. The performance of the models were recorded. All clients that the models identified to be at risk of long-term shelter use were removed from the subsequent window size test sets (normalised 120, and 180 days test sets).

The same 60days decision rules are then used to make predictions on the remaining normalised window size 120days test set. The performance of the models were recorded. All clients that the models identified to be at risk of long-term shelter use were removed from the normalised window size 180days test set. Finally the same 60days decision rules are then used to make predictions on the remaining normalised window size 180days test set. The performance of the models were recorded. The plots of all the scores are represented in Figures 5.9, 5.10 and 5.11. The total number of clients in the test set was 1368.

N = 100				
Window Size	Number of LT	Number of notLT	TP	FP
60	10	90	2	1
90	8	89	3	4
120	5	85	0	1
180	5	84	1	1

Table 5.3: Hypothetical real time program delivery experiment representation

For illustration, lets consider Table 5.3 with hypothetical test sets. The total number of clients in the test sets are 100 (thus; N = 100), with 10 of them been long-term shelter users and 90 of them not long-term shelter users. The decision rule generated from the train set is used to make predictions on the widow size 60 days test set. This decision rule identified 3 clients to be at risk of long-term shelter use. Two of the identified clients are true (TP) and one is false (FP). These 3 identified clients are removed from the widow size 90, 120 and 180 days test sets and referred to human counsellors. This leaves the widow size 90 test set with 8 long-term shelter users and 89 not long-term shelter users as shown in the table.

The same decision rule is then used again to make predictions on the widow size 90 days test set. The rule identified 7 clients to be at risk of long-term shelter use. Three of the identified clients are true (TP) while four are false (FP). These 7 identified clients are removed from the widow size 120 and 180 days test sets and referred to human counsellors. This leaves the widow size 120 test set with 5 long-term shelter users and 85 not long-term shelter users.

		CART	N = 1368			
Window Size	Number of LT	Number of notLT	TP	FP	TN	FN
60	645	723	383	233	490	262
90	262	490	11	1	489	251
120	251	489	6	0	489	245
180	245	489	9	0	489	236

Table 5.4: CART's truth values

		GOSDT	N = 1368			
Window Size	Number of LT	Number of notLT	TP	FP	TN	FN
60	645	723	278	138	585	367
90	367	585	16	2	583	351
120	351	583	7	0	583	344
180	344	583	12	1	582	332

Table 5.5: GOSDT's truth values

		EXPLORE	N = 1368			
Window Size	Number of LT	Number of notLT	TP	FP	TN	FN
60	645	723	225	86	637	420
90	420	637	12	1	636	408
120	408	636	6	1	635	402
180	402	635	13	0	635	389

Table 5.6: EXPLORE's truth values

For the third time the decision rule is again used to make predictions on the widow size 90 days test set. The decision rule identified 1 client to be at risk of long-term shelter use. This identified client is false (FP). The identified client is removed from the widow size 180 days test set and referred to human counsellors. This leaves the widow size 180 test set with 5 long-term shelter users and 84 not long-term shelter users. Finally the decision rule is again used to make predictions on the widow size 180 days test set. The decision rule identified 2 long-term shelter users. These 2 identified long-term shelter users are removed and referred to human counsellors.

Figures 5.10 and 5.11, shows that rules generated from a window size of 60 days data is only effective at making predictions on a clients 60 days of interaction data records. Though Figure 5.9 shows a continuous increase in precision, the truth Tables 5.4,5.5 and 5.6, shows that the algorithms captured considerably high number of true positives within the 60 days window size test set and subsequently capture very low positives in the 90, 120 and 180 days window size test sets. As discussed in Section 5.4, precision is not reliant on the number of long-term shelter users captured but the fraction of referrals who will become long-term shelter users. Though the true positives captured are very low, they are barely capturing an false positive which offers precision a lending hand.

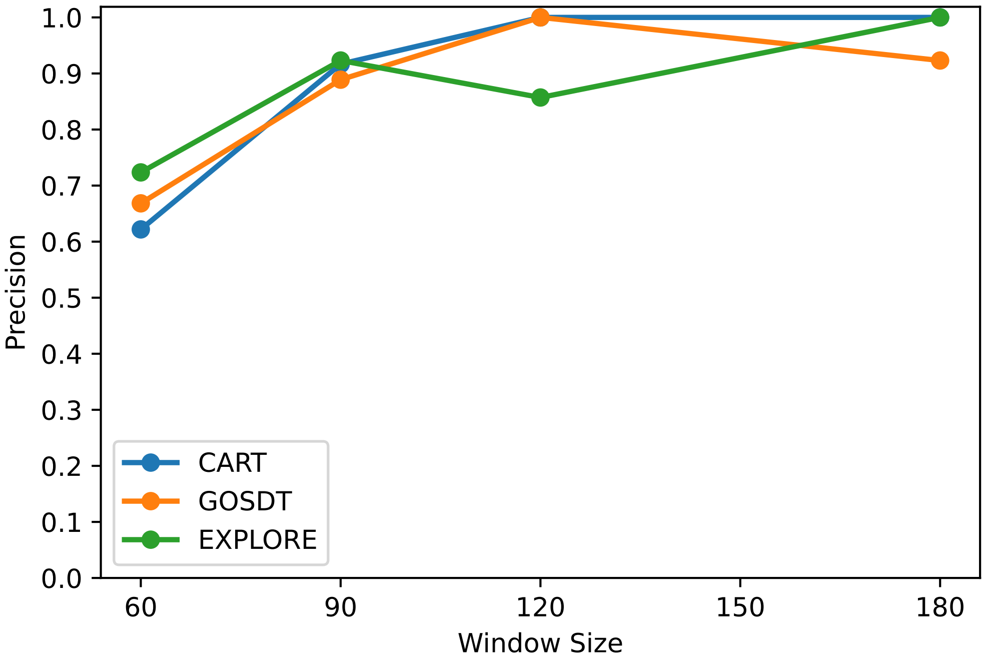

Figure 5.9: Precision of window size 60days' rules applied on all window sizes of the all youth cohort.

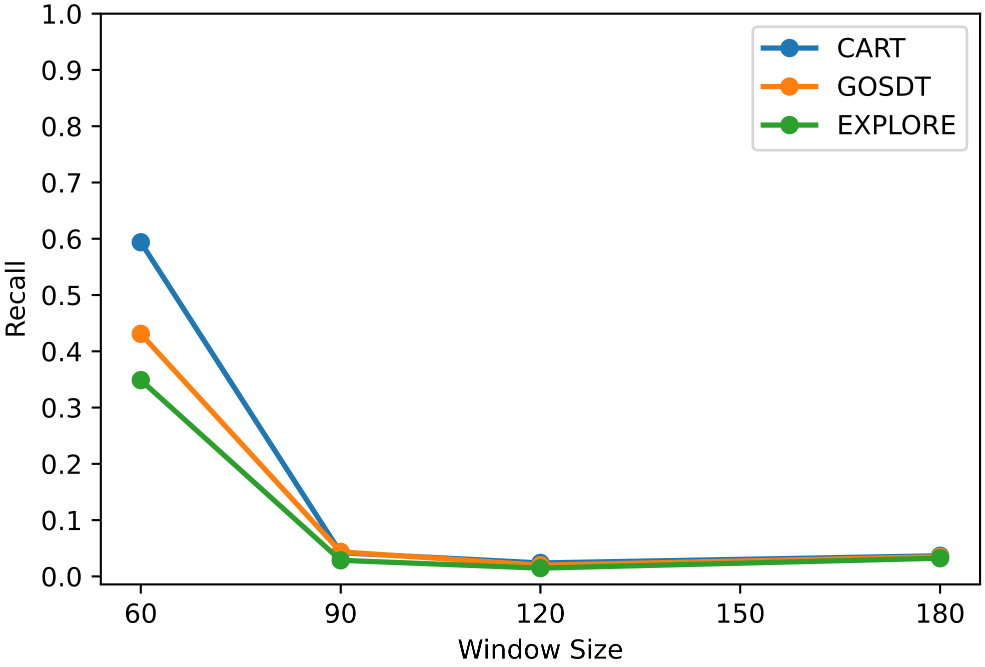

Figure 5.10: Recall of window size 60days' rules applied on all window sizes of the all youth cohort.

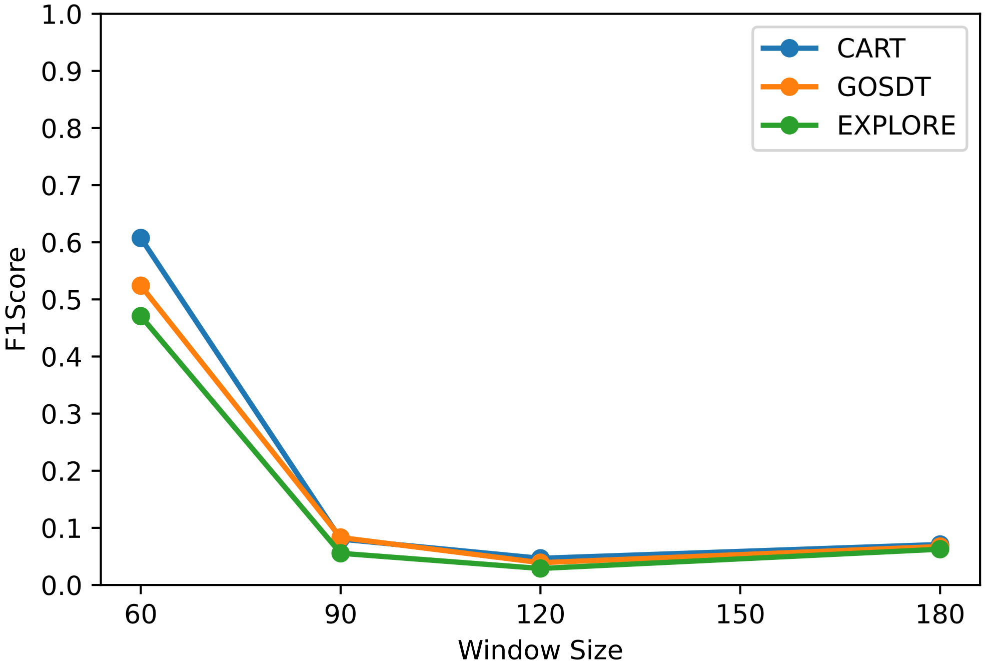

Figure 5.11: F_1Score of window size 60days' rules applied on all window sizes of the all youth cohort.

Conversely, in Figures 5.10 and 5.11 the performance of the models on the 90 days test set and beyond is close to zero. This is due to the fact that, the models are recording a wide gap between true positives and false negatives as can be seen in the truth Tables 5.4, 5.5 and 5.6. As discussed in Section 5.4, recall is the fraction of long term clients captured. Considering few true positives captured as mentioned in the previous paragraph, coupled with high false negatives in the 90, 120 and 180 days window size test set it is understandable that Figures 5.10 and 5.11 are showing very low recall values for all the models with their predictions in the 90, 120 and 180 days window size test sets.

In general the results show that, to effectively make predictions at regular intervals, each interval needs a model trained on records of clients interactions with the DI center equal to that interval's duration. Multiple models needs to be trained and the appropriate model for the appropriate interval used in making predictions.

5.6 Generalization Capacity

Generalization refers to the model's capability to adapt and react properly to new data. In other words, generalization examines how well a model can digest new data and make correct predictions after getting trained. Generalisation was investigated in two ways as follows;

1. **Disruptive period generalization:** Considering the fact that there have been significant policy changes and events that might have affected the way clients interact with the emergency shelter system, how well decision rules generated from one cohort performed in a different cohort was investigated. This is to give an indication, whether decision rules generated with data before a disruption in the patterns of client interactions with the shelter system are still viable decision rules to be applied on a new data after the disruption. This will inform, whether the model needs to always be retrained or not, every time there is a disruption that affects shelter interactions.

2. **Demographic generalization:** The effectiveness of decision rules generated from the general population for identifying youth at risk of long-term shelter use. Are these decision rules sufficient or is there the need to generate tailored decision rules for identifying youth at risk of long-term shelter use?

5.6.1 Disruptive Period Generalization

Figures 5.12, 5.13 and 5.14 shows how rules generated with housing ready youth cohort data, performed on the housing ready and housing first cohorts. In Section 2.3.1, a comprehensive discussion of the idea of cohorts (e.g. housing ready and housing first) including the date ranges have been provided. As described in Section 5.2.2, the housing ready youth window size 90 days cohort data was split in 70% train set and

30% test set. The housing first youth window size 90 days cohort data was treated as a second test set. The algorithms were trained on the same housing ready train set and the generated models used to make predictions on the housing ready youth test set and the housing first youth cohort. CART and GOSDT were trained using a maximum tree height of 2, additionally GOSDT regularization was set to 0.1 while EXPLORE was trained using rule length of 3. The performance of the models on the housing ready test set was plotted on the x-axis against the performance of the models on the housing first cohort on the y-axis.

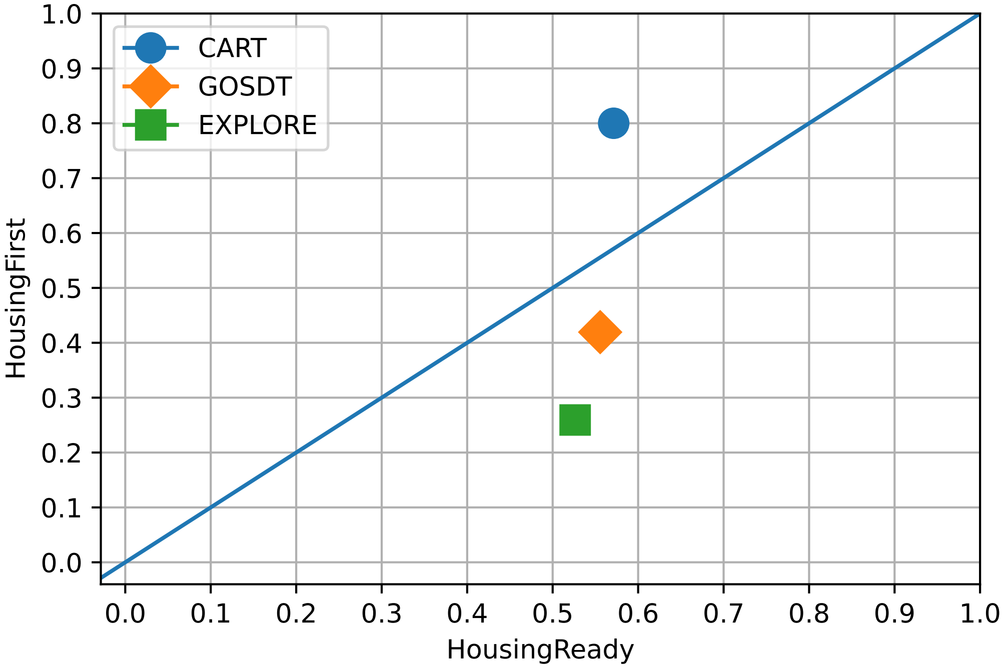

Figure 5.12: Precision of housing ready rules applied on housing ready VS housing first cohorts

If the performance of an algorithm's decision rule falls below the blue line (slope = 1) then it is performing better on the housing ready youth cohort than on the housing first youth cohort. On the other hand, if the performance of an algorithm's decision decision rule falls above the blue line then it is performing better on the housing first youth cohort than on the housing ready youth cohort. If the performance of an algorithm's rule falls on the blue line then it is performing comparably on both the housing ready youth cohort and the housing first youth cohort. If the performance of an algorithm's decision rule falls on or close to the blue line, then the model is deemed to generalise well. To what distance a performance value can be considered close to the line of slope = 1 is subjective. A distance of 0.1 shows that the model is performing 10% less in one dataset. The fact that the population under consideration is a vulnerable group makes the 10% less performance even more pronounced because missing a single long-term client is detrimental. However, there

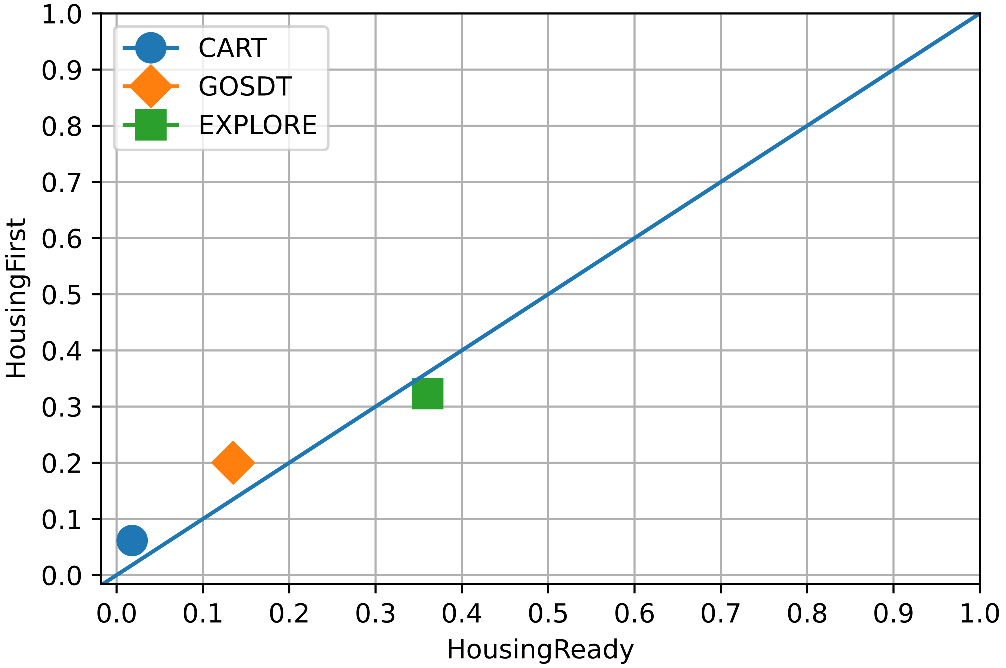

Figure 5.13: Recall of housing ready rules applied on housing ready VS housing first cohorts

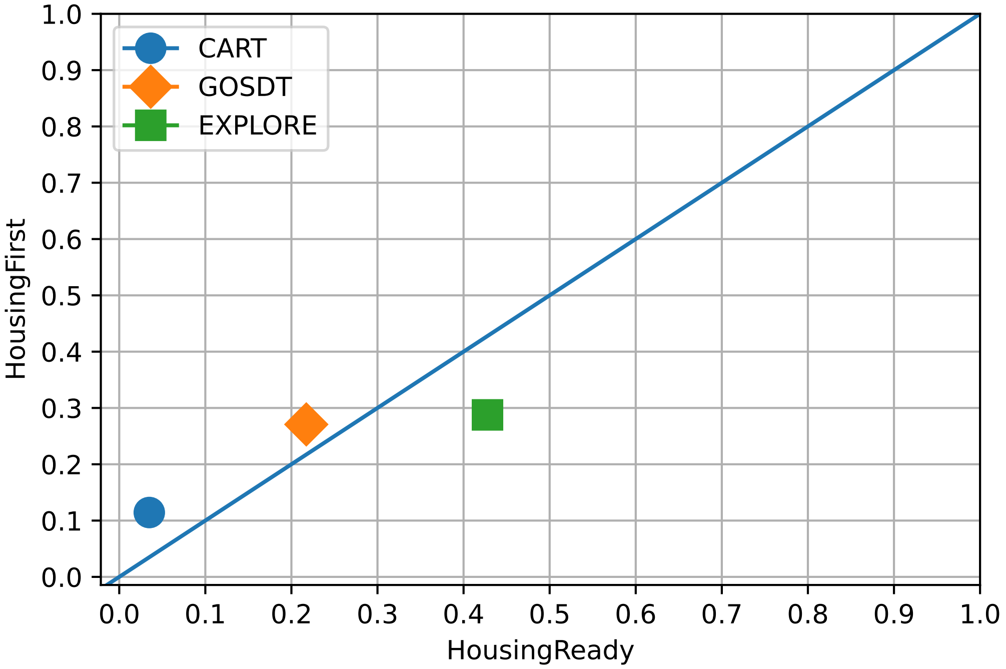

Figure 5.14: F_1Score of housing ready rules applied on housing ready VS housing first cohorts

is an inevitable certain threshold that one must set: For this work a model is considered to generalise if the distance of its performance value to the blue line is less than or equal to 0.1.

Generally, Figures 5.12, 5.13 and 5.14 shows that the models did generalise well. This is an indication that after any significant policy change or event that affects the client interactions with the emergency shelter system, models can still be used to make predictions in the new data set. However, the poor recall and F_1Score performance could be attributed to the imbalance nature of the datasets used for this experiment. As can be seen in Table 2.5 the proportion of long-term clients in the housing ready youth data and the housing first youth data are 32% and 10% respectively.

5.6.2 Demographic Generalization

The performance of detecting youth at risk of long-term shelter use using decision rules generated from all clients data compared to decision rules generated from all youth cohort data is shown in Table 5.7. The all youth 90 days window size cohort data and the all clients 90 days window size data were split into disjoint sets as described in Section 5.2.2. The algorithms were trained on the all youth cohort train set and the all client train set, generating two decision rules each; all youth decision rule and all client decision rule. CART and GOSDT were trained using a maximum tree height of 2, additionally GOSDT regularization was set to 0.1 while EXPLORE was trained using rule length of 3. The generated decision rules were then used to make predictions on the all youth test set and the algorithms scored.

	All Youth Rule			**All Clients Rule**		
Algorithm	Precision	Recall	F_1Score	Precision	Recall	F_1Score
CART	0.699	0.4357	0.5368	0.6814	0.4775	0.5615
GOSDT	0.6635	0.5349	0.5923	0.6814	0.4775	0.5615
EXPLORE	0.7273	0.4093	0.5238	0.7083	0.369	0.4852

Table 5.7: Performance of all clients rules and all youth rules applied on all youth cohort

In general the values in Table 5.7, shows that the all youth decision rules of all the algorithms are performing slightly better than the all clients decision rules, at predicting youth at risk of long-term shelter use. However, this difference in performance is marginal. As discussed in Section 5.6.1, the difference is less than 0.1. It can therefore be inferred that decision rules generated with all clients dataset can effectively be used for the task of predicting long-term youth shelter users.

5.7 Generated Rules

Table 5.8 presents the DNF decision rules that were generated by all the algorithms in the real-time program delivery experiment. The models were trained on the all youth window size 60 days train set, as indicated in Section 5.5. The table also shows how well these decision rules performed when used to make predictions on the all youth window size 60 days test set.

Algorithm	**Rule**	**Recall**	**Precision**
CART	EventSleep > 9.0	0.5938	0.6218
GOSDT	EventSleep ≥ 24	0.4310	0.6683
EXPLORE	EventBar $> 0.0 \;\vee\;$ EventSleep > 18.0	0.4636	0.6689

Table 5.8: Real time program delivery rules

The best performing DNF decision rules produced by all the algorithms when trained on the all youth window size 90 days train set, are also presented in Table 5.9. Tree heights of 2, 3, and 4 were used to train CART and GOSDT. GOSDT was additionally constraint with a regularisation of 0.1 for more sparse models. EXPLORE was trained with rule lengths of 1, 2, 3,4 and 5. The table shows how well these decision rules performed when used to make predictions on the all youth window size 90 days test set.

Algorithm	**Rule**	**Recall**	**Precision**
CART	EventSleep $> 5.5 \;\wedge\;$ EventBuildingCheckIn $\leqslant 0.5$	0.3256	0.7167
GOSDT	EventSleep ≥ 3	0.4822	0.6561
EXPLORE	EventBuildingCheckIn $\leqslant 1.0 \;\wedge\;$ EventSleep > 4.0	0.3659	0.7195

Table 5.9: Decision rules from the all youth cohort

The decision rules presented in both tables are very simple and easy to understand, can be implemented on any simple spread sheet or database.

5.8 Summary

Several programs (including street outreach, coordinated access and assessment, as well as predictive tools) are relied upon to identify and connect chronically homeless people with housing resources. To avoid overfitting, disjoint data samples were created for training and testing the models. Stratified sampling was applied to generate these disjointed datasets. To ensure fairness, the algorithms were trained and tested on the same dataset subsets. The decision trees were pruned in order to reduce complexity while enhancing performance. For the experiments that entailed training the models on one observation window and making predictions on another, the data was normalised. Despite the fact that pruning the decision trees reduced the complexities of the resulting models, CART and GOSDT decision rules still grew significantly as the decision tree's height increased. Training and predicting on both the youth and adult cohorts revealed that machine learning alone is insufficient in the fight against homelessness, but it is extremely useful in identifying at-risk long-term shelter clients very early in their homelessness. Less complicated models retain comparable predictive power while being more interpretable. Making predictions at regular intervals for real-time program delivery necessitates the employment of models trained on records of clients' interactions with the DI centre equal to the duration of that interval. Decision rules generated from all clients can be used to effectively predict long-term youth shelter users. Finally, the DNF rules produced by the algorithms are readable and implementable by a non-technical person.

Chapter 6

Conclusion

The purpose of this chapter is to reiterate the research problem, sum up the journey to the findings presented in the thesis, wrap up ideas discussed in the research, summarise the key findings and also suggest opportunities for future research. The rest of this chapter is organised as follows; a recap of the study is presented in Section 6.1 and proposals for future work is discussed in Section 6.2.

6.1 Summary

Homelessness is a peculiar condition that exposes an individual to a vast range of life adversities beginning with, lose of property, mental and physical health deterioration, substance abuse, sexual violence, discrimination and the list continues endlessly. Youth long-term shelter users are notably the most vulnerable within this vulnerable population. The importance of preventing the condition from happening can not be over emphasised and the data science community have a vital role to play. Data science solutions however must be tools to augment the expertise of humans, hence should be interpretable, easy to implement and cost effective.

Data is a key component in machine learning. It is the most important factor that enables algorithm training and explains why machine learning has gained so much popularity in recent years. However, no matter how much data is provided, if it is not presented properly, a machine will be essentially worthless. The data was therefore pre-processed into a form that epitomises the problem at hand. To achieve this the data was segregated into youths and adults, into the different significant event periods that may have an impact on the records and also into time windows.

Classification rules in DNF format is a fundamental requirement for this research and only interpretable algorithms that possess the capability of ultimately producing classification rules in DNF format were investigated. However some of the algorithms were tree models and therefore, the decision trees needed to be translated into decision rules in DNF format. In Section 3.5 the method used to translate decision trees into classification rules in DNF format have been described.

Three algorithms were investigated; CART, GOSDT and EXPLORE. CART which is a decision tree model, remains a foundational model for interpretability and is often called upon when inherent interpretability is sought. Unlike most decision tree algorithms, GOSDT is a novel decision tree model that promises optimal sparse decision trees over a range of objectives functions . EXPLORE, a novel algorithm that employs exhaustive search to induce classification rules in DNF form, allowing for user defined performance constraints and giving direct control of the complexity of the DNF rules.

It was shown in Section 5.2 that CART and GOSDT do not offer direct control of the complexity of their classification rules in DNF format. The DNF rule complexity of CART and GOSDT increased rapidly with increasing tree height. Though both pre-pruning and post pruning of the CART and GOSDT decision trees helped reduce the complexity of the decision trees considerably, the output decision rules in DNF format were not reflective of the decision tree heights.

Increased complexity do not lend outstanding predictive power over simpler models. It has been demonstrated in Section 5.4, that increasing the complexity of a model only returns a marginal increase in performance. Hence the benefit of having a simpler model out-ways the marginal increase in performance that the extra complexity brings.

Rules generated from one era can still be used in predicting long-term shelter users during another era. Significant policy changes and major event disruptions that may alter the shelter access patterns do not affect the models considerably. Hence new decision rules do not necessarily need to always be generated after pivotal disruptions impact the shelter interactions. This is evident in the results discussed in Section 5.6.

It has been shown in Section 5.5, that the most viable option for a real time program delivery solution is to generate multiple models trained on different intervals of data records and the appropriate model for the appropriate interval used in making predictions. This is a demonstration of implementing machine learning solutions incorporating real time program delivery which does not call for operational changes at the shelter system.

Decision rules generated from all clients' data records can be utilised to identify youth who are at risk of long-term emergency shelter use. This was clearly demonstrated in Section 5.6.

6.2 Limitations

There are limitations, like with every research work. The limitations that follow should be considered when interpreting the results.

First and foremost, the study's dataset is limited, consisting of records of interactions of homeless clients with the Calgary drop in centre and does not extend to cover every individual experiencing homelessness in Calgary. A portion of the homeless population does not use emergency shelter services. According to a report [61], nearly a third of Arizona's homeless population sleeps rough and does not use emergency shelters. As a result, this group of homeless individuals are not included in this study and will not benefit from its findings. Collecting data from rough sleepers and integrating it with data from emergency shelters to form a larger dataset could go a long way towards helping this sub-population to benefit from machine learning solutions.

Secondly, the findings of this study cannot be applied to other cities or nations. Because emergency shelters in other cities in Canada and even other nations operate differently. The characteristics of the data collected by these emergency shelters in other cities or countries would most likely differ from the dataset obtained by the Calgary drop-in centre. Different cities' datasets may have less or more features than the dataset utilised in this study. A specific feature of the dataset on which a rule is based may be missing in a dataset from a different city. Furthermore, because emergency shelters in different cities define data features differently, the implications of the values of data features in different cities' emergency shelters data will differ. If emergency shelter operations and data collection are streamlined across Canada, the findings of research like this will be generalizable across the country.

6.3 Future Work

Every research work presents new pathways to extending the work further for the betterment of humanity. This work is not an exception. The author has some few suggestion he feels are worth pursuing. These are listed as follows in no particular order.

The current research, lumps all predictions together as long-term shelter users and not long-term shelter users, irrespective of how far into the future this occurrence is likely to happen. It would be interesting to model the data in such a way that a machine learning algorithm can learn from the data and be able to predict when a client will meet the long-term shelter use criterion. Considering the fact that housing resources are not limitless and individual experiencing homelessness are of different levels of need. This would allow rapid housing programs to prioritize individuals with the most severe service needs for housing support. This will not just be a tool to aid in identifying long-term shelter users but also a tool to help with efficient resource allocation.

There are always going to be individuals that will fall short of any definition set out for chronic homelessness or long-term shelter users. But some of these people still makes good candidates for housing support. It will be worthwhile for future predictive models to assign risk scores to predictions. Assigning risk scores to predictions will allow shelter staff to give individuals with high risk scores a closer attention and could even be referred for housing support. Most often than not a person's priority for permanent housing support goes up if the person is identified to be at risk of becoming chronically homeless. Similarly, if a person has a high risk score and housing resources are available, this person who may have been overlooked because he or she is not chronically homeless and was not identified to be at risk of long term shelter use could suddenly be evaluated for housing assistance.

Administrative data from the police and emergency medical services could be used in predicting chronic homelessness. The present administrative data from the emergency shelters utilised to predict chronic homelessness, as stated in Section 6.2, has a short fall. Not all people experiencing homelessness use these shelters for a variety of reasons. Anybody experiencing homelessness but does not access shelter services may never be identified for housing support by a machine learning solution. Interaction with the police and emergency medical services are mostly involuntary, therefore a machine learning algorithm could unearth hidden useful patterns that leads to chronic homelessness for some individuals that ordinarily would not have been identified for housing support just because they did not access the emergency shelter services. Merging these datasets with the emergency shelter dataset could be a more efficient approach.

It would be interesting to investigate less interpretable algorithms in order to benefit from their powerful prediction ability. Less interpretable machine learning algorithms are often thought to be more discriminatory than more interpretable algorithms. However, because this work placed a high value on interpretability, implementability, model complexity, and complexity control, algorithms that did not meet these criteria were not examined. In contrast to this work, which took a deterministic approach, less interpretable algorithms such as Naïve Bayes (a probabilistic algorithm) could be highly fascinating to investigate.

Finally but not the least, the problem of homelessness could also be modeled as a multi-class problem. The current work in machine learning space in the context of homelessness mostly predicts whether an individual will be chronically homeless some time into the future or not. But there are other groups within the homelessness space like episodic homelessness and transitional homelessness that also needs attention. A machine learning algorithm may be able to learn better predictive patterns of episodic or transitional homelessness than chronic homelessness. Identifying these other homelessness is half of the solution.

Bibliography

[1] S. Fazel, J. R. Geddes, and M. Kushel, “The health of homeless people in high-income countries: descriptive epidemiology, health consequences, and clinical and policy recommendations,” *The Lancet*, vol. 384, pp. 1529–1540, Oct. 2014.

[2] S. Gaetz, E. Dej, and T. Richter, *Homelessness Canada in the State of 2016.* Toronto, ON, CA: Canadian Observatory on Homelessness Press, 2016. OCLC: 1000914534.

[3] U. N. (March 2023), “The human right to adequate housing https://www.ohchr.org/en/special-procedures/sr-housing/human-right-adequate-housing.”

[4] Employment and S. D. C. (March 2023), “Homelessness data snapshot: The National Shelter Study 2019 update https://www.infrastructure.gc.ca/homelessness-sans-abri/reports-rapports/data-shelter-2019-donnees-refuge-eng.html.”

[5] S. A. Gaetz, O. Bill, S. Kidd, and K. Schwan, *Without a home: the National Youth homelessness survey.* Toronto [Ontario]: Canadian Observatory on Homelessness Press, 2016. OCLC: 980600374.

[6] L. B. Whitbeck, D. R. Hoyt, and W.-N. Bao, “Depressive symptoms and co-occurring depressive symptoms, substance abuse, and conduct problems among runaway and homeless adolescents,” *Child development*, vol. 71, no. 3, pp. 721–732, 2000.

[7] K. A. Yoder, “Comparing suicide attempters, suicide ideators, and nonsuicidal homeless and runaway adolescents,” *Suicide and Life-Threatening Behavior*, vol. 29, no. 1, pp. 25–36, 1999.

[8] M. Shaw and D. Dorling, “Mortality among street youth in the uk,” *The Lancet*, vol. 352, no. 9129, p. 743, 1998.

[9] E. Roy, N. Haley, P. Leclerc, B. Sochanski, J.-F. Boudreau, and J.-F. Boivin, “Mortality in a cohort of street youth in montreal,” *JAMA : the journal of the American Medical Association*, vol. 292, pp. 569–74, 09 2004.

[10] P. N. Goering, D. L. Streiner, C. Adair, T. Aubry, J. Barker, J. Distasio, S. W. Hwang, J. Komaroff, E. Latimer, J. Somers, and D. M. Zabkiewicz, "The At Home/Chez Soi trial protocol: a pragmatic, multi-site, randomised controlled trial of a Housing First intervention for homeless individuals with mental illness in five Canadian cities," *BMJ Open*, vol. 1, pp. e000323–e000323, Nov. 2011.

[11] Employment and S. D. C. (March 2023), "Reaching Home: Canada's Homelessness Strategy https://www.infrastructure.gc.ca/homelessness-sans-abri/directives-eng.html."

[12] N. H. Busen and J. C. Engebretson, "Facilitating risk reduction among homeless and street-involved youth," *Journal of the American Academy of Nurse Practitioners*, vol. 20, pp. 567–575, Nov. 2008.

[13] C. Pitcher, E. Saewyc, A. Browne, and P. Rodney, "Access to Primary Health Care Services for Youth Experiencing Homelessness: "You shouldn't need a health card to be healthy."," *Witness: The Canadian Journal of Critical Nursing Discourse*, vol. 1, pp. 73–92, Dec. 2019.

[14] S. A. Gaetz, *The state of homelessness in Canada 2013.* Toronto, Ont.: Homeless Hub, 2013. OCLC: 858604278.

[15] S. Pomeroy, "The cost of homelessness: Analysis of alternate responses in four canadian cities."

[16] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "THE COMPUTATIONAL LIMITS OF DEEP LEARNING,"

[17] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the Computational Cost of Deep Learning Models," in *2018 IEEE International Conference on Big Data (Big Data)*, (Seattle, WA, USA), pp. 3873–3882, IEEE, Dec. 2018.

[18] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges," July 2021. arXiv:2103.11251 [cs, stat].

[19] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A Survey Of Methods For Explaining Black Box Models," June 2018. arXiv:1802.01933 [cs].

[20] H. Hagras, "Toward Human-Understandable, Explainable AI," *Computer*, vol. 51, pp. 28–36, Sept. 2018.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," Aug. 2016. arXiv:1602.04938 [cs, stat].

[22] D. Gunning and D. W. Aha, "DARPA's Explainable Artificial Intelligence Program,"

[23] J. Liu, C. Zhong, M. Seltzer, and C. Rudin, "Fast Sparse Classification for Generalized Linear and Additive Models," p. 30.

[24] J. Lin, C. Zhong, D. Hu, C. Rudin, and M. Seltzer, "Generalized and Scalable Optimal Sparse Decision Trees," Aug. 2020. arXiv:2006.08690 [cs, stat].

[25] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin, "Learning Certifiably Optimal Rule Lists," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (Halifax NS Canada), pp. 35–44, ACM, Aug. 2017.

[26] P. R. Rijnbeek and J. A. Kors, "Finding a short and accurate decision rule in disjunctive normal form by exhaustive search," *Machine Learning*, vol. 80, pp. 33–62, July 2010.

[27] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," Sept. 2019. arXiv:1811.10154 [cs, stat].

[28] J. Tan, B. Arch, and M. Eng, "Using Machine Learning to Identify Populations at High Risk for Eviction as an Indicator of Homelessness,"

[29] A. Malik, "Predicting Chronic Homelessness at an Emergency Homeless Shelter in Calgary using Neural Network models and Time-Stamped data records,"

[30] C. Chelmis, W. Qi, W. Lee, and S. Duncan, "Smart Homelessness Service Provision with Machine Learning," *Procedia Computer Science*, vol. 185, pp. 9–18, 2021.

[31] A. Kube, S. Das, and P. J. Fowler, "Allocating Interventions Based on Predicted Outcomes: A Case Study on Homelessness Services," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 622–629, July 2019.

[32] T. Byrne, A. E. Montgomery, and J. D. Fargo, "Predictive modeling of housing instability and homelessness in the Veterans Health Administration," *Health Services Research*, vol. 54, pp. 75–85, Feb. 2019.

[33] B. Hong, A. Malik, J. Lundquist, I. Bellach, and C. E. Kontokosta, "Applications of Machine Learning Methods to Predict Readmission and Length-of-Stay for Homeless Families: The Case of Win Shelters in New York City," *Journal of Technology in Human Services*, vol. 36, pp. 89–104, Jan. 2018.

[34] A. R. Kube, "Fair and Efficient Allocation of Scarce Resources Based on Predicted Outcomes: Implications for Homeless Service Delivery,"

[35] H. Chan, E. Rice, P. Vayanos, M. Tambe, and M. Morton, "Evidence From the Past: AI Decision Aids to Improve Housing Systems for Homeless Youth,"

[36] B. VanBerlo, M. A. S. Ross, J. Rivard, and R. Booker, "Interpretable Machine Learning Approaches to Prediction of Chronic Homelessness," Sept. 2020. arXiv:2009.09072 [cs].

[37] G. G. Messier, L. Tutty, and C. John, "The Best Thresholds for Rapid Identification of Episodic and Chronic Homeless Shelter Use," May 2021. arXiv:2105.01042 [cs].

[38] G. G. Messier, C. John, and A. Malik, "Predicting Chronic Homelessness: The Importance of Comparing Algorithms using Client Histories," May 2021. arXiv:2105.15080 [cs].

[39] C. John and G. G. Messier, "A Rule Search Framework for the Early Identification of Chronic Emergency Homeless Shelter Clients," May 2022. arXiv:2205.09883 [cs].

[40] G. I. Webb, "OPUS: An Efficient Admissible Algorithm for Unordered Search," *Journal of Artificial Intelligence Research*, vol. 3, pp. 431–465, Dec. 1995.

[41] E. K. Jr, "Early Intervention to Prevent Persistent Homelessness,"

[42] C. H. F. (August 2023), "https://www.calgaryhomeless.com/discover-learn/our-approach/housing-first-people-first/."

[43] S. GAETZ, F. SCOTT, and T. GULLIVER, *Housing First in Canada.* Toronto, ON, CA: Homeless Hub, 2013. OCLC: 1374507614.

[44] G. of Canada (July 2023), "https://www.infrastructure.gc.ca/homelessness-sans-abri/index-eng.html."

[45] G. of Alberta (July 2023), "https://open.alberta.ca/dataset/7d9d2a8e-1392-4ab8-a4bb-6ba4aac6b285/resource/bc702fc0-a5ac-4ddf-a92f-98adae7af22c/download/planforab-secretariat-final.pdf."

[46] C. D. C. (July 2023), "https://calgarydropin.ca/wp-content/uploads/2021/04/Calgary-Drop-In-Centre-2018-19-Report-to-Community-website.pdf."

[47] T. Byrne and D. P. Culhane, "Testing Alternative Definitions of Chronic Homelessness," *Psychiatric Services*, vol. 66, pp. 996–999, Sept. 2015.

[48] P. Flach, *Machine Learning: The Art and Science of Algorithms that Make Sense of Data.*

[49] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees."

[50] E. Allender, L. Hellerstein, P. Mc, and T. Pitassi, "MINIMIZING DNF FORMULAS AND AC0 CIRCUITS GIVEN A TRUTH TABLE,"

[51] S. Khot and R. Saket, "Hardness of Minimizing and Learning DNF Expressions," in *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, (Philadelphia, PA, USA), pp. 231–240, IEEE, Oct. 2008.

[52] C. Umans, "The Minimum Equivalent DNF Problem and Shortest Implicants," *Journal of Computer and System Sciences*, vol. 63, pp. 597–611, Dec. 2001.

[53] X. Hu, C. Rudin, and M. Seltzer, "Optimal Sparse Decision Trees," Sept. 2020. arXiv:1904.12847 [cs, stat].

[54] S. Nijssen and E. Fromont, "Mining optimal decision trees from itemset lattices," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, (San Jose California USA), pp. 530–539, ACM, Aug. 2007.

[55] R. Galan and S. Gambino, "Beyond normality: the predictive value and efficiency of medical diagnosis," in *:*, J. Wiley & Sons, 1975.

[56] N. Slesnick, X. Feng, X. Guo, B. Brakenhoff, J. Carmona, A. Murnan, S. Cash, and A.-L. McRee, "A Test of Outreach and Drop-in Linkage Versus Shelter Linkage for Connecting Homeless Youth to Services," *Prevention Science*, vol. 17, pp. 450–460, May 2016.

[57] T. P. O'Toole, E. E. Johnson, M. L. Borgia, and J. Rose, "Tailoring Outreach Efforts to Increase Primary Care Use Among Homeless Veterans: Results of a Randomized Controlled Trial," *Journal of General Internal Medicine*, vol. 30, pp. 886–898, July 2015.

[58] C. H. F. (September 2023), "https://www.calgaryhomeless.com/discover-learn/our-approach/system-planning/coordinated-access-assessment/."

[59] H. Liu, A. Gegov, and M. Cocea, "Complexity Control in Rule Based Models for Classification in Machine Learning Context," in *Advances in Computational Intelligence Systems* (P. Angelov, A. Gegov, C. Jayne, and Q. Shen, eds.), vol. 513, pp. 125–143, Cham: Springer International Publishing, 2017. Series Title: Advances in Intelligent Systems and Computing.

[60] H. Liu, M. Cocea, and A. Gegov, *Interpretability of Computational Models for Sentiment Analysis*, pp. 199–220. Cham: Springer International Publishing, 2016.

[61] L. Larsen, E. Poortinga, and D. E. Hurdle, “Sleeping Rough: Exploring the Differences Between Shelter-Using and Non-Shelter-Using Homeless Individuals,” *Environment and Behavior*, vol. 36, pp. 578–591, July 2004.

Appendix A

Metrics

A.1 Background

The study's evaluation of the machine learning algorithms is crucial. When measured against one metric, such as accuracy, a model might produce satisfactory results, but when measured against other metrics, it might produce unsatisfactory results. Although accuracy is frequently used to gauge a model's performance, it is insufficient for this task to thoroughly evaluate the models. The various forms of statistical evaluation measures used are covered in this section.

A.2 Confusion Matrix

Let's look at the confusion matrix, which can assist in comprehending the various metrics. A confusion matrix is a matrix representation of the prediction summary. It displays the number of correct and wrong predictions for each class. It aids in comprehending the classes that are confused by the model as another classes. A confusion matrix for a binary classification task is shown in Table A.1.

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

Table A.1: Confusion matrix

Based on the truth values of the actual and anticipated classes, there are four possible outcomes, as indicated in Table A.1:

- **True Positive (TP)** — An example has been predicted to be positive and it is really positive.

- **True Negative (TN)** — An example has been predicted to be negative and it is really negative.
- **False Positive (FP)** — An example has been predicted to be positive but it is actually negative.
- **False Negative (FN)** — An example has been predicted to be negative but it is actually positive.

A.3 Accuracy

Accuracy represents the percentage of right predictions out of all predictions. It is the number of correct predictions divided by the total number of predictions.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy is the most often used measure since it is very straightforward and simple to understand. While straightforward, the usefulness of accuracy score is largely dependent on data specifics. The result will be untrustworthy if the dataset is imbalanced (the classes in a set are presented unevenly). For example, in a dataset with 98 percent samples of class A and only 2 percent samples of class B . By just predicting all examples as class A, the model can easily achieve 98 percent accuracy.

A.4 Precision

Precision indicates what percentage of all positive predictions were right. To compute it, the number of correct positive results (TP) is divided by the total number of positive (TP + FP) predictions made by the classifier.

$$Precision = \frac{Number\ of\ correctly\ predicted\ positives}{Total\ number\ of\ predicted\ positives} = \frac{TP}{TP + FP}$$

Precision is ideal, when there is a need to avoid false negatives but cannot overlook false positives. However, it is not an universal solution because false negatives and actual negatives must be considered in some circumstances. For example, it is critical to understand how many true long-term shelter users were identified as transitional and did not receive housing assistance.

A.5 Recall

Recall represents the proportion of correct positive predictions out of all positives. It is the total number of true positives divided by the total number of true positives and false negatives.

$$Recall = \frac{Number\ of\ correctly\ predicted\ positives}{Total\ number\ of\ positives} = \frac{TP}{TP + FN}$$

Unlike the precision metric, recall signals missed positive predictions. In this study's problem, it answers the question, "How many long-term shelter users did the model correctly predict?" It is vital to discover all long term shelter users from a humanitarian standpoint, hence it is fine if the model identifies some transitional shelter users as long term shelter users. They will almost certainly take some permanent housing resources earmarked for long term shelter users, which is unacceptable but not critical. However, it is far worse if the model classifies some long-term shelter users as transitional shelter users and does not offer them with the chance of getting housing assistance. In this case, recall outperforms precision because it increases the number of long-term shelter users who are correctly predicted for housing assistance.

A.6 F1 Score

Precision and recall are usually in conflict. In other words, improving precision often results to a decrease in recall and vice versa. Figure A.1, known as precision-recall (PR) curve, shows this concept clearly. Predicting all examples as positives results in a recall of one, but the precision is quite low due to the large amount of false positives. On the other hand, if just the most likely samples are predicted to be positive, precision increases since the number of true and predicted positives are close. However, recall is low this time. Precision and recall should ideally be kept high.

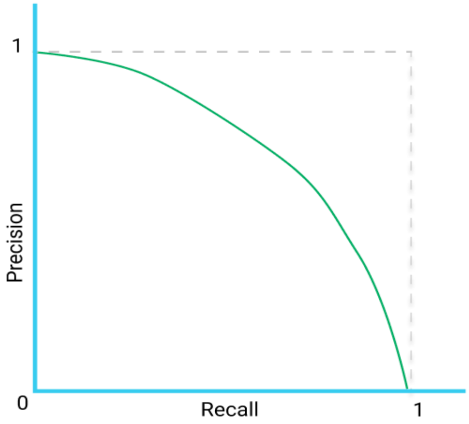

Figure A.1: Precision – Recall Curve

The F1 score is a measure that attempts to harmonise precision-recall pairings. By computing their harmonic mean, the F1 Score attempts to establish a balance between precision and recall. A maximum value of 1, suggests that the precision and recall are perfect.

$$F_1\ Score = Harmonic\ mean\ of\ precision\ and\ recall = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

Balancing precision and recall by merely taking the average of the two opens the door to false prediction accuracy. The F1 Score is a more elaborate metric that produces more realistic results on imbalanced problems. In contrast to a high F1 Score, a low F1 Score is not as informative: it merely indicates the performance at a threshold without indicating whether it is a recall or precision error.