ELDERLY FALLS PREDICTION AND PREVENTION USING MACHINE LEARNING

Submitted by

Nyashadzashe Nziramasanga
Master of Science (Computer Science)
FAN: nzir0001, Student ID: 2243700
nzir0001@flinders.edu.au

Supervisor

Dr. Trent Lewis, trent.lewis@flinders.edu.au

Submitted to Flinders University – Adelaide Australia College of Science and Engineering for the partial fulfilment of the degree of Master of Science (Computer Science)

Submission Date

20 October 2021

DECLARATION OF ORIGINALITY

I Nyashadzashe Nziramasanga, a Master of Science (Computer Science) student researcher, acknowledge that this thesis:

1. Does not incorporate any work without acknowledging any resources and material previously submitted for a degree or diploma at any other tertiary institution.
2. To the best of my knowledge, the thesis does not contain any material previously published by another individual except where due reference is made.

Signature: [redacted]

Print Name: Nyashadzashe Nziramasanga

Date: Wednesday, 20 October 2021

ACKNOWLEDGMENTS

I would like to thank my principal supervisor, Dr Trent Lewis, for the invaluable assistance, providing guidance and insight into artificial intelligence and machine learning techniques during the research. I would also like to recognise the effort and support my co-supervisor, Dr David Hobbs, for establishing communication with AnglicareSA management.

My sincere thanks go to the management team at AnglicareSA for being accommodating, providing the datasets and other relevant material for the research effort. Special thanks to Mr Kristian Bennet and Mrs Jenna Falzon, the primary contacts at AnglicareSA, for helping to navigate the project, assisting with their guidance, domain expertise and providing access to the Anglicare system and materials.

Finally, I wish to thank my family and friends for their constant praise and support throughout my research. This journey would not have been possible without the financial and emotional support I received during the journey.

Author's contributions:

Dr Trent Lewis (Senior Lecturer at Flinders University College of Science and Engineering)
trent.lewis@flinders.edu.au

Dr David Hobbs (Senior Lecturer at Flinders University College of Science and Engineering)
david.hobbs@flinders.edu.au

Anglicare

[redacted]

DISCLOSURE

The thesis was conducted in partnership with AnglicareSA, a non-profit organisation based in Adelaide, Australia. The datasets presented and information relating to AnglicareSA are proprietary and of commercial interest to AnglicareSA. Due to privacy and security concerns, the AnglicareSA clinical health information and elderly falls reports datasets are not redistributable to third parties other than those engaged in the study.

ABSTRACT

Background

Advances in computational processing and the availability of large datasets in healthcare presents an opportunity to apply machine learning (ML) techniques to aid in predicting patterns in clinical datasets, assisting in diagnosing and treating patients.

The thesis examines the application of three ML algorithms on aged care datasets to predict the severity of an elderly fall. The thesis also aims to investigate the appropriate means of monitoring, collecting data, and analysing the likelihood of elderly falls to reduce the costs related to elderly falls and the severity of falls in aged care facilities.

Method

Using falls incident reports and clinical information datasets sourced from AnglicareSA, a not-for-profit aged care and social services provider. Three ML algorithms were built for the research, namely Decision Tree Classifiers (DTC), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). The models were compared against each other based on their accuracy, precision, recall and F-score to classify the severity of fall incidents and build a fall prediction model supporting clinical reasoning.

Results

About 2187 falls incidents and clinical data records remained after pre-processing. The mean age was 79, with most fall incidents were reported happening in bedrooms with minor outcomes for the severity of the fall. The accuracy for DTC, MLP and SVM was moderate at best recorded as 60%, 69% and 39%, respectively. The top five features that contributed significantly to predicting the severity of falls were incident location, age, number of incidents, facility, and respiratory rate.

Conclusion

Even though the study explored the use of DTC, MLP and SVM algorithms to classify the severity of falls based on the recorded falls incidents and clinical health information, with reasonable prediction accuracy. However, future work is required to improve the accuracy of the ML models by using larger datasets of elderly falls and clinical datasets and applying wearable devices to help predict a fall.

Keywords: Decision Tree Classifier, Support Vector Machine, Multi-Layer Perceptron, Predictive modelling, Machine learning and Elderly falls

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ABBREVIATIONS	DESCRIPTIONS
ADCC	Aged, Disability & Community Care
ADL	Activities of daily living
AI	Artificial Intelligence
AIHW	Australian Institute of Health and Welfare
AUC	Area Under Curve
AUC	Area Under the Curve
BMI	Body Mass Index
BPM	Breaths Per Minute
CDC	Centres for Disease Control and Prevention
DTC	Decision Trees Classifier
DTC	Decision Tree Classifier
FN	False Negative
FP	False Positive
GCS	Glasgow Coma Score
K-NN	K-nearest neighbour
LOF	Local Outlier Factor
LR	Logistic Regression
MELMV	Multi-view Ensemble Learning with Missing Values
ML	Machine Learning
ML	Machine Learning
MLP	Multi-layer Perceptron
NHMD	National Hospital Morbidity Database
NN	Neural Network
RFC	Random Forest Classifier
RFC	Random Forest Classifier
RL	Reinforcement Learning
SA	South Australia
SL	Supervised Learning
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
USA	United States of America
WA	Western Australia
WHO	World Health Organisation

1. INTRODUCTION

In recent years, advances in healthcare technology and computational processing have enabled Artificial Intelligence (AI) to apply to the vast sums of electronic health records available. Learning from the immense amount of healthcare data available in the digital age help find patterns, correlations and making sense of data. The application of ML techniques can be applied for pattern recognition and predictive analysis. Well-developed models demonstrate diagnostic acumen that surpasses human capabilities and at scale (Mark Sendak, 2019).

Falls are a significant threat to the quality of life for elders (Phelan E. A., 2015) and are most prevalent at advanced ages. Each year, approximately 30% to 40% of people aged 65 years and older living within the community will experience a fall incident (Mary E. Tinetti, 1988). Roughly half of all falls result in an injury (Mary B. King, 1995), and 10% of falls lead to severe injuries such as hip, thigh, pelvic, head fractures or even deaths.

Impacts and costs related to falls

Falls have been a growing issue worldwide, where falls result in injuries and deaths from complications related to falls. According to a 2021 report by the World Health Organisation (WHO), falls are the second leading cause of injury and death worldwide (World Health Organisation, 2021). The death rates are highest amongst adults over 60 years of age. WHO estimates that over 684 000 individuals die due to falls and fall-related injury globally each year and that 37.3 million reported falls cases are severe enough for individuals to require medical attention each year.

The United States of America (USA) recorded a total of 37,587 deaths in 2017 because of falls, and this accounted for 15.5% of all injury deaths according to the Centers for Disease Control and Prevention (CDC) 2019 National Vital Statistics Report (Centers for Disease Control and Prevention, 2019). Elderly falls accounted for over half the falls and 85% of the deaths. According to the CDC, falls are the leading causes of death, injury, and hospital admissions among the elderly population. Trends show that non-fatal falls and fall-related injuries are prevalent amongst adults aged 65+ years (Briana Moreland, 2020).

The trends in increasing hospitalised injury due to falls in older people 2007-08 to 2016-17 (Australian Institute of Health and Welfare and Flinders University, 2019) report by the Australian Institute of Health and Welfare (AIHW), using data from the National Hospital Morbidity Database (NHMD). Found that about 125,000 people aged 65 and over were hospitalised due to falls in 2016-17—the cause of falls was from slipping, tripping, or stumbling on the same level. Around 85% of the fall-related cases in 2016-17 were recorded as occurring in homes or aged care facilities. The estimated recurrent health service expenditure for rehabilitation and treatment on fall-related injuries cost $3.9 billion nationally (AIHW, 2019).

In 2000 fall-related costs in the USA reached $20 billion dollars for both fatal and non-fatal falls (J A Stevens, 2006). Fall-related injuries and treatment among older adults are

associated with substantial economic costs such as rehabilitation; meaning falls prevention is necessary to minimise the financial and emotional costs occurring from elderly falls. A 2012 study on Incidence and costs of injury in Western Australia (WA) falls where found to account for 32.5% of fatal injuries, 32.5% of non-fatal hospitalisations, and over A$2.2 billion in costs (Hendrie D, 2012).

AnglicareSA Background

AnglicareSA has partnered with Flinders University College of Science and Engineering to assist in researching and developing a prediction and prevention of elderly falls Machine Learning model to monitor elderly residents prone to falls.

Listed below are AnglicareSA's main objectives:

- Applying machine learning techniques to pre-emptively predict and alert elderly falls risk.
- A roadmap to ensure that appropriate monitoring, data collection and analysis is put into place that can guide improved practice to reduce the number and severity of falls.

Considerations and Challenges

Due to the recency and sensitive nature of the research, the following challenges listed below explain some of the constraints, considerations and challenges experienced throughout the research, which should be factored in for.

- The study is new and conducted with a large external organisation AnglicareSA. Some challenges experienced were sourcing the data from multiple systems within AnglicareSA, enrolling and training to understand the internal workings and processes of AnglicareSA, and requesting setting up a centralised database with all data from each system.
- Data entered in RiskMan and iCareHealth was not entered with the prospect of being used for ML research, and thus the datasets contained formatting and other issues that are detrimental to the task of ML. The Datasets needed extensive data cleaning, such as merging datasets collected from two separate systems (RiskMan and iCareHealth). Other processing steps taken include removing null values and producing derived values. For example, with weight and height, a BMI score can be derived same with age can be derived from the date of birth (DOB). Unfortunately, most prediction models cannot handle missing values in datasets, which must be addressed prior to modelling.
- The motivation is to use clinical explainable ML models because the requirement by AnglicareSA clinicians is to interpret the outcome of every decision made by the prediction model.

Causes of Elderly Falls

There are multiple factors why elderly falls occur (Sollitto, 2021); below are some reasons contributing to elderly falls.

- Decline in physical fitness at an advanced age, where most adults become less active at older ages. Failure to engage in minor physical activity leads to the degradation of muscle mass, strength, and reduced bone mass, resulting in poor balance.
- Impaired vision occurs because of age-related eye diseases, making it incredibly difficult for elders to detect fall hazards such as uneven ground or steps.
- Environmental hazardous factors are another contributor to falls at residential aged care homes, hazards such as poor lighting, loose carpets, or lack of safety equipment such as guide rails, ramps, and lifts impact elderly safety.
- Surgical operations such as hip replacements can leave elderly individuals weak or less mobile than before the procedure. Rehabilitation of elderly individuals is crucial

in assisting them to recover from past operations and helping them quickly recover their physical ability as soon as possible.

- Side effects from medication due to an increase in the wide variety of medicines prescribed to elders in their advanced ages, with some side effects being drowsiness or dizziness from sedatives, antidepressants, or opioids. A study observing polypharmacy among adults aged 65 years and older in the United States from 1988 to 2010 found that 39% of adults aged 65 and overtook more than five medications for chronic health conditions (Christina J. Charlesworth, 2015)

2. REVIEW OF LITERATURE

Researchers have captured, and classified elderly falls in various methods from wearables devices, vision-based systems, incident reporting and clinical health records. The performance of an ML model has traditionally been measured based on the accuracy of the ML model to identify and classify a fall correctly. However, a clear emerging trend emphasises the use of multiple hybrid approaches for capturing falls data and using the vast sums of data for scalable fall prediction and prevention systems.

This section presents a substantial review of related work and background research on predicting elderly falls. Also, investigating the existing data capture methodologies and how ML models classify falls in fall prediction and prevention systems—a look into the outcomes of the methodologies and examining any potential gaps in their literature. As literature around elderly falls is widely documented, a focus shall be placed on the gaps in the research and how they were overcome. Next, the techniques used in different works of literature shall be compared against each other to highlight the strengths and weaknesses realised from each of the studies.

2.1 Elderly Falls Studies

Falls have accounted for a significant number of hospitalisations and long-term care admission in older adults (Faulkner, 2007). Most of the elderly falls were caused by navigating steps and ambulation at home and age care facilities. Various research has been carried out around preventing elderly falls in different contexts; for the literature review, the context shall focus on ML methodologies and hardware used to capture information for the ML models.

Data collection techniques

For ML models to make accurate predictions, vast sums of data are required, meaning the essential stage is acquiring the data. Data can be acquired from digital forms filled by clinicians or information from a sensor on a device. How the data is retrieved drastically affects the outcome of the ML model prediction. Even though in this study the datasets were provided by AnglicareSA, a core component of the research is to recommend appropriate data capture mechanisms to assist in predicting elderly falls more accurately.

There have been multiple solutions proposed for elderly falls detection data capture methodologies and can be categorised into three main areas, namely wearable devices, non-wearable devices, and a hybrid of the systems (Diana Yacchirema, 2019). Most non-wearable devices use vision-based devices such as cameras to detect falls and have proven to be extremely effective. One study used a deep learning technique called R-CNN, which performs scene analysis and gauges the relationship of human figures and furniture in the space to detect a fall occurring. The fall detection system reported performance of 94.44% precision, 94.95% recall and 95.50% accuracy (Weidong Min, 2018). Another study used a Microsoft Kinect sensor to detect falls using depth sensors on the Kinect camera (Mastorakis, 2014). Though vision-based devices have proven to be effective in detecting

elderly falls, the main drawback is the high cost of purchasing all the necessary resources to capture and process the video feed. Also, the lack of privacy for elderly individuals as the devices would need to be installed in nearly all indoor environments and constantly powered on to be deemed effective enough to detect falls occurring.

Figure 1 illustrates a vision-based fall detection system that utilises a Microsoft Kinect sensor to capture the activity of an elderly resident and classifies their activity by training an SVM algorithm to predict a fall (R. Alazrai, 2015).

Figure 1. Anatomical-plane-based human activity representation for elderly falls detection (R. Alazrai, 2015)

When it comes to wearable devices, some studies have had relative successes with the use of commonly available smartphone devices with their built-in accelerometers and gyroscopes. Smartphones are able to continuously monitor and track the movement of elderly people (Diana Yacchirema, 2019). A study proposed the application of an activity

recognition system for elderly individuals using their smartphones (Miguel Ángel Álvarez de la Concepción, 2017). However, Pasqui's (G. Plasqui, 2013) concluded that the best position for wearing an accelerometer device to track daily activity was close to the centre of mass, hence the lower back or hip of an elderly individual.

It is evident that wearable devices are the most effective method of capturing data to detect and predict falls. The downside is the cost of the wearable devices and the need for constant charging to remain useful whilst in use.

A popular approach for fall detection is using a hybrid solution of both wearable and non-wearable devices to capture data from multiple points and validate them against each other. Casilari demonstrates that by using data from an android personal device (smartphone) and a smartwatch accelerometer and gyroscope, he was able to distinguish a fall from the normal daily activities of patients (Casilari, 2015).

The primary data collection at AnglicareSA when a fall occurred was through a digital form being filled out on RiskMan. The form is a falls incident form which is presented in Figure 7, where a nurse, clinician or staff member would fill out after an elderly fall occurred. Some issues have arisen from missing information in the data and lack of accuracy on the exact time and location a fall could have occurred.

Overall, the study relied on sourcing data from human entered digital forms after a fall's incident occurred. A more effective means of data collection would have been using multiple wearable devices and non-wearable devices such as a smartphone and a smartwatch.

2.2 Datasets, Data types and Features

The features of a dataset matter in understanding what can be classified and used for an ML prediction model. This section shall examine some key data features which contribute to the outcome of a prediction.

Andrew theorised that ML algorithms could accurately predict the mortality after a fall compared with the standard logistic regression (LR) model based on immediately available admission data (Andrew J. Young, 2021). The study included 4725 patients admitted for a fall-related injury between 2012 and 2017 with ages of 20 to 61 years old, with average hospital stays of 5 to 7 days. The experiment was to predict who would be discharged from the hospital and determine which variables had the most significant effect on the prediction. The experiment revealed that Glasgow Coma Score (GCS) motor, GCS verbal, respiratory rate, GCS eye and temperature were the five variables that contributed most to the prediction of mortality in descending order of priority. Figure 2 below illustrates the features that contributed significantly to the prediction of mortality.

Figure 2. Feature Importance (Andrew J. Young, 2021)

Specific motor, verbal and respiratory rate features contribute significantly more to the final prediction of an elderly fall.

David and colleagues to predict the severity of inpatient fall using an ML classifier multi-view ensemble learning and model-based missing data imputation methods (David S. Lindberg, 2020). The study used over two thousand inpatient falls data sourced from the HMH clinical data warehouse (proprietary data), which consisted of patient's demographic characteristics (age, sex, and race), diagnoses, bone density measurements and procedural data. The multi-view ensemble learning with missing values (MELMV) dealt with multi-source patient data and achieved a cross-validated AUC of 0.713 (95% CI, 0.701–0.725). The MELMV classifier produced a severe fall index used to identify patients most at risk of severe injuries if they fall, allowing proper clinical intervention to prevent a fall.

2.3 Machine Learning Classification Techniques

Researchers use several ML classification techniques for detecting and predicting falls. Decision Tree Classifiers (DTC), Random Forest Classifiers (RFC), Support-vector Machine (SVM), Artificial Neural Networks (ANN) and Logistic Regression (LR) are some of the most popular techniques being applied in this space.

This section will briefly examine ML classification techniques widely used in related studies around falls detection and prediction ML models applied. The emphasis is on comparing the performance of each ML model and evaluating its suitability for research.

A study by Aziz (Aziz, 2017) revealed that SVM was the best at classifying accuracy after comparing and evaluating five different ML algorithms, namely logistic regression, decision tree classifier, K-nearest neighbour, Naïve Bayes and SVMs. The three measures of performance the algorithms were evaluated against were the false-positive rate, specificity,

and sensitivity. Aziz experimented using a waist-mounted triaxial accelerometer on ten young volunteers, and the SVM algorithm achieved a performance rate on sensitivity and specificity of 96% and 96%, respectively, when identifying the of falls from the activity of daily living. However, a gap in the research was the small group of young adults used to simulate falls and that by validating the performance using accuracy, specificity and sensitivity are not sufficient when the goal is to reduce a long lie (time on the ground after a fall has occurred).

A similar study carried out by Speiser (Jaime Lynn Speiser, 2021) discusses the use of DTC and RFC algorithms and proposes a method for developing a prediction model for serious fall injury using a from Lifestyle Interventions and Independence for Elders (LIFE) study. The study found that the performance of DTC and RF was moderate at best, with an area under the receiver operating curve (AUC) of 54% for DTC and 66% for RF. Nevertheless, the LIFE dataset used for the study comprised of adults at high risk of falling and involvement in the study may have prevented severe falls from occurring.

Yang (S. Yang, 2013) proposes a falls prediction algorithm (FPA), using a wearable device with an inertial sensor package called an inertial measurement unit (IMU), shown in Figure 3. The experiment showed that the FPA could predict a fall 0.4 seconds prior to the beginning of a fall and a performance of 70%. The FPA algorithm adopted a neural network to predict a fall occurring, and the inputs were the accelerations and angular rates of the upper trunk, whilst the outputs were “fall” or “no fall”. Though this is a novel approach to detecting and predicting falls, the study aim is to pre-emptively deploy a safety mechanism such as an inflatable airbag to reduce injuries due to falls.

Figure 3. Inertial Measurement Unit – IMU (S. Yang, 2013)

Pillai (Pillai, 2020) developed a fall detection system that classifies daily activities into either fall or non-fall actions. The study used an open-source SisFall dataset which consists of Gait data collected using a Tri-axial accelerometer (Angela Sucerquia, 2017). The data is collected using an ADXL34 accelerometer strapped around the participant’s waist, as illustrated in Figure 4. The triaxial accelerometer ADXL345 is used to get the acceleration values along three axes, x, y and z and is energy efficient as it can work with low power mode. Two ML

algorithms, SVM and DTC were compared against each other, and the findings were accuracy 84.17%, training time 294.95 seconds and prediction time 84.71 seconds for SVM and accuracy of 95.87%, training time 2.741 seconds and prediction time 0.02 seconds for DTC.

Figure 4. Waist belt with sensor position (Pillai, 2020)

Another study proposed by Santoyo-Ramón (Santoyo-Ramón, 2018) compared the performance of SVM, K-NN, DTC and Naive Bayes algorithms using a wearable Fall Detection System (FDS). The author used four sensors located on different body positions such as the waist, ankle, chest and thigh, and all sensor data were sent to a smartphone app that distinguished different ADLs amongst falls. The outcome of the research revealed that the chest and waist were the most suitable locations for the sensors to improve the effectiveness and precision of the system. Another finding by Santoyo-Ramón was that the SVM algorithm and sensor placed on the waist was able to achieve a sensitivity and specificity higher than 93%. Nevertheless, the study produced notable results; the smartphone was a notable point of failure due to all the processing requirements from the data sent from the sensors, which quickly drained the smartphone's battery.

A notable research by Wang (Y. Wang, 2017) proposes a device-free fall detection system named WiFall which can detect a fall without additional hardware or wearable devices. WiFall uses existing wireless infrastructure such as access points, and 802.11n NIC equipped desktops, as illustrated in Figure 5. The system uses anomaly detection with Local Outlier Factor (LOF), defined as the ratio of average local densities of one's object neighbour to the local density of the object (Y. Wang, 2017). The WiFall dataset employed an SVM classifier and ensemble Random Forest algorithm. The SVM classifier yielded a 90 % fall detection precision and Random Forest score of 94% fall detection precision, meaning the Random Forest performed 4 % better than SVMs. However, both the WiFall algorithms had a false alarm rate above 12% and that the system only experimented on one person.

Figure 5. WiFall signal propagation model indoor environment (Y. Wang, 2017)

The literature review has identified that firstly wearable sensors with accelerometers and gyroscopes, incredibly close to the centre of mass such as the waist or chest, capture more accurate gait data. Though the study uses data sourced from the AnglicareSA system, the recommendation for more accurate data would be to apply a combination of wearable and non-wearable devices as a tool for fall prediction and prevention. Another observation from the review showed that SVMs, DTC and ANN algorithms yielded reprimandable results, which will be explored further in Section 3.

3. RESEARCH METHODOLOGIES

As mentioned in section 1, the research aims to apply machine learning techniques to datasets provided by AnglicareSA and experiment with DTC, SVM and MLP classifiers to predict the severity of an elderly fall, also known as the outcome from a fall.

3.1 Tools and software

Table 1. lists and briefly describes the software and resources utilised during the research while developing the ML models.

Table 1. Tools and software used for research

Tool	Description
Python	A general-purpose programming language and popular choice for machine learning (Unpingco, 2019)
Scikit-learn	It is a free and open-source machine learning software for python, providing various machine learning models for classification, regression, and clustering. Some algorithms include support vector machines and random forests.
Visual Studio Code	Also known as VS Code is a code editor created by Microsoft
Jupyter notebook	A free and open-source, interactive web tool where documents are formatted based on JSON created by project Jupyter.
Mito	A python data analysis package that helps explore, transform, and present data like Microsoft excel sheets. Features include creating graphs, data exploration, column filtering, data frame merging, pivot tables and spreadsheet formulas.
Pandas	Data analysis and manipulation tool for python data frames
NumPy	Scientific computing library with multi-dimensional arrays
Matplotlib	Library for creating static, animated and interactive visualisations.

Figure 6. presents the procedures taken to develop each ML model, which is discussed in more detail throughout Section 3. In brief, Figure 6. Illustrates that the data was sourced from two systems used by AnglicareSA, namely Telstra iCareHealth and RiskMan. Telstra iCareHealth is software where clinical health information for each resident is entered by clinical staff, nurses, and care workers on a periodical basis. RiskMan is an internal incident reporting and risk management tool used by staff members to enter fall incident reports when they occur. The data from RiskMan and Telstra iCareHealth is exported then stored on a Microsoft Azure Data Lake as unstructured NoSQL data, which the IT department at AnglicareSA set up to support the research as a centralised database did not exist prior to this project.

Once all the data was uploaded to the Microsoft Azure Data Lake at AnglicareSA, a data export was provided of the fall's incidents and clinical health information. The exported datasets went through rigorous data evaluation to understand what information was

available for the prediction model. Part of the data evaluation process was consulting the domain experts to understand the data collection workflow better. A further set of feature selection was conducted, identifying the most suitable features/variables for the research then extensive data preparation was applied to the datasets. The data preparation merged the falls incidents and clinical health records, which have a one-to-many relationship. This was followed by removing missing values, rows, columns, and label encoding the string values into categorical and numeric values, which work well in ML models.

Once the data preparation phase was complete, the resultant dataset was experimented on DTC, SVM and MLP models. All the findings and results were recorded. The rest of section 3 goes into more detail on procedures carried out for each stage of the development pipeline.

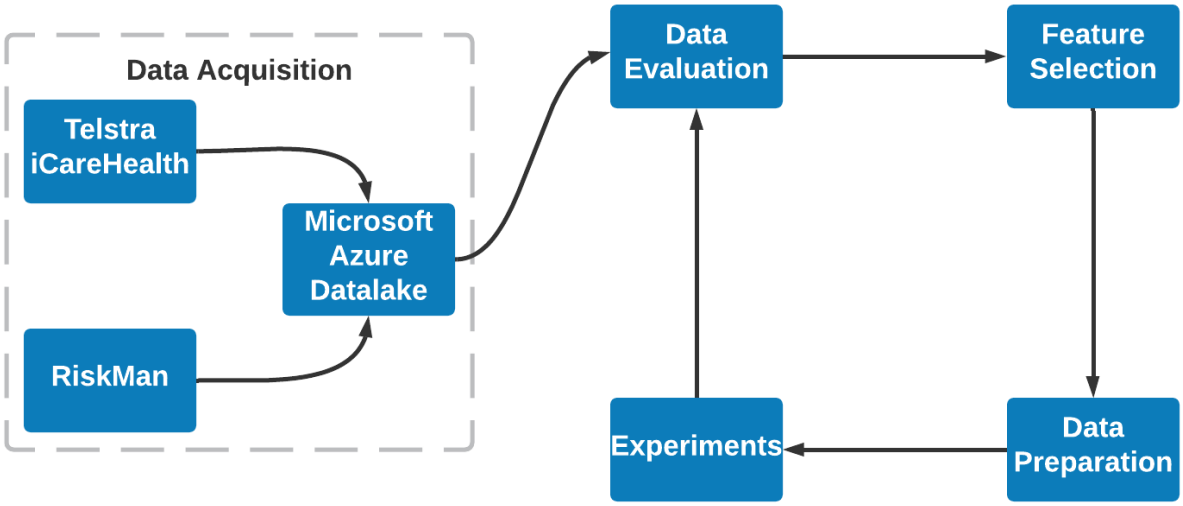

Figure 6. ML model development pipeline

3.2 Data Acquisition

The study uses fall incident reports and clinical health data from AnglicareSA systems and is provided as ".csv" and ".xlsx" files. The datasets consisted of 4,265 falls incident reports and 11,808 clinical assessments reports from February 2019 to Aug 2021, which is one year worth of data. The information was entered by clinicians, nurses, physiotherapists, occupational therapists, care workers, and admin staff conducting clinical assessments and reporting any falls incidents as the residential facilities. Note that the data was not entered for use with ML algorithms.

Data extracted from AnglicareSA's iCareHealth and RiskMan systems were stored together on a Microsoft Azure Data Lake (Microsoft, 2021) as unstructured data for evaluation and data analysis.

iCareHealth is a Telstra Health (Telstra, 2021) product recently renamed Aged, Disability & Community Care (ADCC), a clinical, medication, and workforce management software for residential aged care, disability, and community care providers. AnglicareSA uses the Telstra iCareHealth platform to capture and retrieve integrated resident and clinical information,

which is synched to Medicare, a publicly-funded scheme for universal healthcare in Australia.

RiskMan (RLDatix, 2021) is an internal incident and feedback reporting tool developed by RLDatix as a patient safety solution for compiling with healthcare standards worldwide. AnglicareSA staff members use RiskMan to report incidents by filling in digital incident report forms, as illustrated in Figure 7.

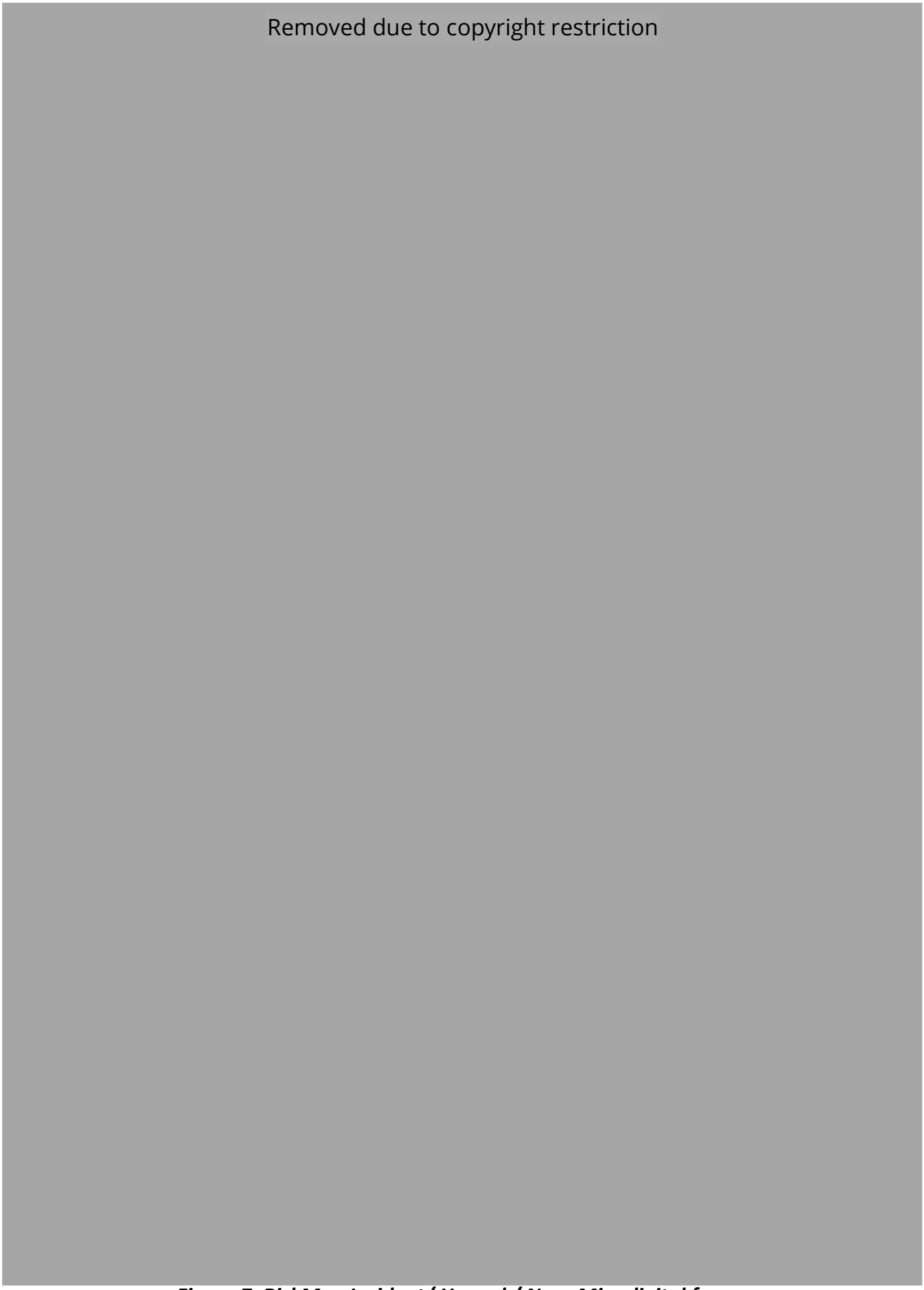

Figure 7. RiskMan Incident/ Hazard / Near Miss digital form

Below are some of the data capture points used by AnglicareSA used to collect falls related information:

- Onsite incident reports entered filed on RiskMan.
- Falls Risk Assessment Tool (FRAT), an online form filled out on the Telstra iCareHealth system.
- Resident clinical information and vital signs measurements include Body Mass Index (BMI), respiratory rate, weight, and height.
- Clinical assessments were recorded by clinical staff members.

3.3 Data Evaluation

This section assesses the data sourced from AnglicareSA and gathers valuable insights based on what was available in the datasets. The provided datasets from AnglicareSA where *clinical_basic_info* and *all_falls_cases*, a sample snapshot of the datasets can be found in Appendix B.

Clinical_basic_info Dataset

The dataset contains 11,808 records collected from each AnglicareSA resident on a periodical basis when they are due for a medical examination. The information is entered through Telstra iCareHealth by medical practitioners and clinicians in digital forms. The data was collected from March 2019 to August 2021, about 2 years and 5 months' worth of medical examinations for the residents.

All_falls_cases Dataset

The dataset contains 4,265 records of reported falls incidents that occurred in AnglicareSA aged care facilities. The information is entered into RiskMan, an AnglicareSA internal risk management and incident reporting tool. The mean age is 79 years for residents reported to have experienced a fall, with most of the severity of the fall being minor and occurring in bedrooms. The data was collected from February 2019 to August 2021, which is about 1 year 6 months' worth of falls incident which has been reported. In Figure 8 the age of the resident's chart shows a good example of data entry issues that occurred where a clinician entered a resident age as 0; for the research, this was removed during pre-processing of the dataset.

Below from Figures 8 to 12 are charts are based on the *clinical_basic_info* and *all_falls_cases* datasets, which present valuable information to understand the distribution, location, and severity of falls.

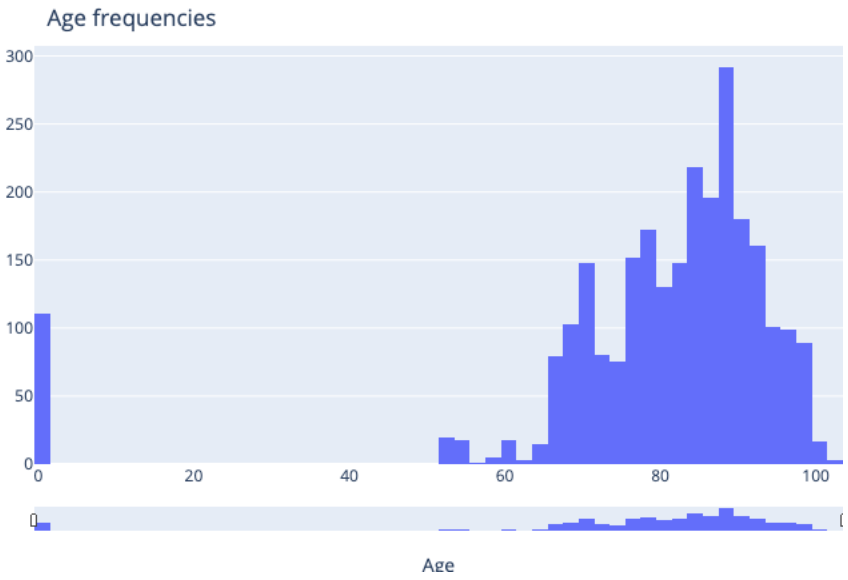

Figure 8. Age of residents

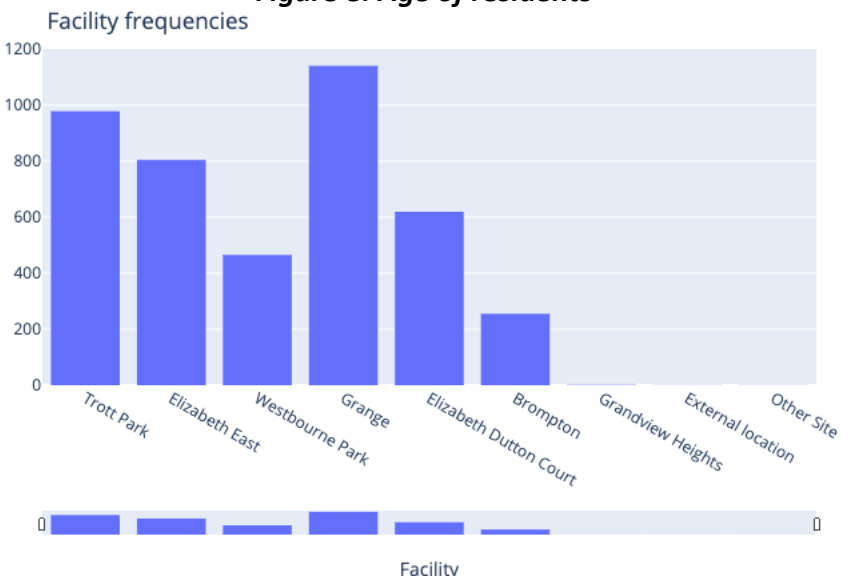

Figure 9. Number of falls per facility

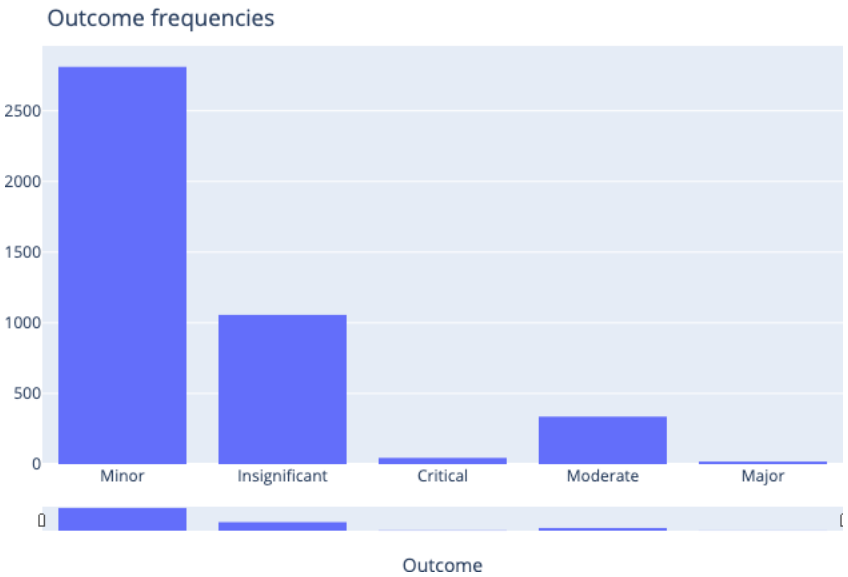

Figure 10. Distribution of severity of falls

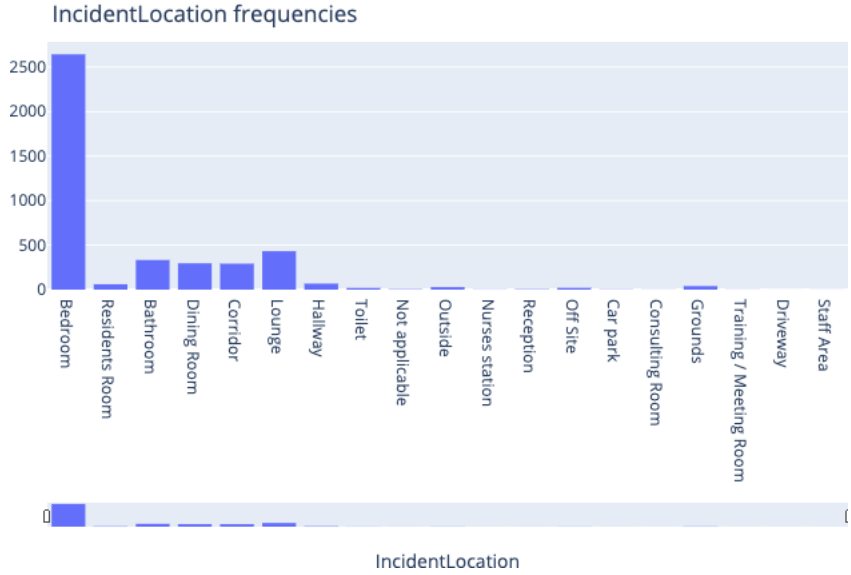

Figure 11. Location of where falls occur

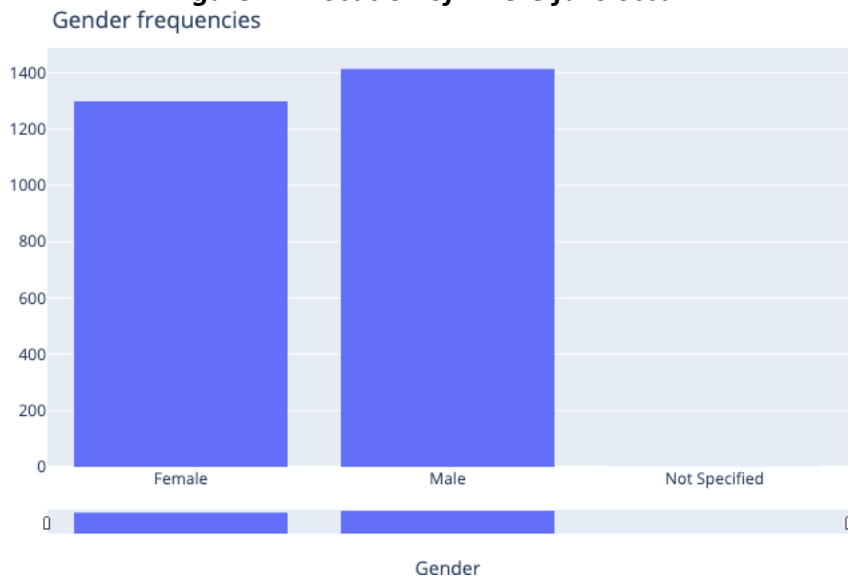

Figure 12. Gender distribution of falls

3.4 Feature Selection

Feature selection is the process of selecting the most relevant and informative features (Guyon I., 2006). The motivation for feature selection is general data reduction, which improves algorithm processing speed, performance improvement to gain a higher prediction accuracy, and knowledge on how the data was generated.

The approach taken was first analysing the features/variables of the datasets to understand the domain and possibilities with the dataset and what features would contribute to a fall's detection and prediction. The features were selected with the aid of expert advice from AnglicareSA nurses, clinical practitioners, and allied health staff. The normalisation of the data was not necessary to avoid breaking relationships within the two datasets.

Table 2 and Table 3 list the features, a brief description of each feature and the data types on the fall's incident report and clinical basic information datasets.

Table 2. Falls Incident Report Dataset

Feature	Description	Data Type
CareReceiverId	Unique resident/patient identification number	Number
DisplayId	Unique falls incident report identification number	Number
Age	Age of the resident at the time of fall	Number
DateOfBirth	Date resident was born	Date/Time
Gender	Male or Female	String
BodyPartAffected	Location of injury on the body	String
AdminssionDate	Date falls was recorded	Date/Time
IncidentDate	Date fall incident occurred	Date/Time
IncidenentTime	Time fall incident occurred	Date/Time
IncidentLocation	Location of the incident	String
Facility	Aged care Facility where fall occurred	String
Facility_NumberOfBeds	Number of people a facility can bed	Number
NumberOfIncidents	Number of fall incidents that have been recorded to date	Number
Outcome	Severity of the fall that occurred	String

Table 3. Clinical Assessment Basic Info Dataset

Feature	Description	Data Type
CareReceiverId	Unique resident/patient identification number	Number
DisplayId	Unique falls incident report identification number	Number
ObserveredDate	Date resident was seen	Date/Time
IncidentDate	Date and time fall incident occurred	Date/Time
Respiratory – rate	Resident rate of breathing, measured in Breaths per Minute (BPM)	Number/ String
Weight	Weight of resident in kilograms (kg)	Number
Height	Height of resident in centimetres (cm)	Number
BMI	Body Mass Index of resident	Number
Activity	Activity resident engages, for example, TV viewing, physical games, reading, prayers etc	String

Based on the datasets, the following features were selected and applied to the ML prediction models:

Falls incident report selected features:

- DisplayID
- Age
- DateOfBirth
- Gender
- BodyPartAffected
- AdmissionDate
- IncidentDate
- IncidentTime
- IncidentLocation
- Facility
- NumberOfIncidents
- Outcome

Clinical Assessment Basic Info features:

- DisplayID
- IncidentDate
- Respiratory – rate
- Weight
- Height
- BMI

3.5 Data Preparation

Data preparation or cleaning refers to all kinds of tasks and activities to detect and repair errors in the dataset (Chu, 2019). The process of cleaning data requires sound knowledge and context of the dataset prior to proceeding. There are multiple types of errors that can exist in a dataset, most commonly missing values, duplicate rows, unnecessary columns. As most of the data were collected through individuals filling in digital forms, there is potential for human errors, such as duplications, incorrect information entered or missing values that all need to be accounted for before passing data through an ML model.

Also, the data was initially collected for a completely different purpose which was to capture health records and report incidents. As the context, the data was collected was different, the assumption that the datasets are not collected for the purpose of using an ML model to predict falls and their outcome. To that end, extensive data pre-processing and cleaning was applied to the provided datasets to make it suitable for the ML models to interpret the existing data and find valuable patterns.

Table 4 identifies the problems observed during the data cleaning procedure and how the dataset was processed.

Table 4. Data cleaning procedures

Issue	Mitigation/ Process
Merging Datasets	The fall cases and clinical basic information dataset were merged using the CareReciverId as the primary key found in both datasets. Three is a many-to-many relationship between the fall's cases clinical basic information dataset, where one fall report maps to many clinical and one clinical record maps to many fall reports. The final dataset meant that all the fall's incidents had the latest clinical information for each resident.
Missing/null values	The datasets contained missing values, which are a common issue when data is collected from digital forms entered by clinical and nursing staff and that the forms were not designed with the context of collecting information related to the fall's prediction and prevention study. The rows and columns with missing values were either filled in with a median value or removed altogether. This is because ML models are not effective when there are missing/null values in a dataset.
Irrelevant fields	Removed columns that were deemed insignificant to contributing to the study. In this instance, for the basic clinical information, dataset, A*ctivity* and *DisplayID* columns were removed after an expert recommendation from clinicians that they would have minimal at best contribution to the study.

4. EXPERIMENTS

4.1 Machine Learning Models

This section compares the performance of three classifiers Decision Tree Classifier (DTC), Support Vector Machine (SVM) and Multi-layer Perceptron (MLP), to predict the severity of a fall based on the pre-processed dataset. Described below, in brief, is the history of machine learning followed by a description of each ML model applied to the research.

4.1.1 Definition and overview of Machine Learning

To understand ML and how it has evolved over the decades, we must first explore the history of ML. The first mathematical model of neural networks, which was published in a scientific paper by Walter Pitts and Warren McCulloch in 1943, was the fundamental foundation for ML (Pitts, 1943). This was followed by Arthur Samuel, a researcher in computer gaming and artificial intelligence at IBM who first coined the term "Machine learning" in 1959 (Samuel, 1959).

Machine learning is considered a subfield of artificial intelligence and comprises several multidisciplinary domains, including computer science, mathematics, statistics, artificial intelligence, data mining and deep learning, data science and natural language processing (Subasi, 2020). Tom M. Mitchell, a computer scientist and professor at Carnegie Mellon University (CMU), defined ML algorithms as "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E*" (T. Mitchell, 1986).

Supervised, unsupervised and reinforcement learning are the three main ML paradigms. Learning paradigms determines the pattern the ML model learns something or from someone. The three paradigms are explained in brief below.

Supervised learning (SL)
Supervised learning (SL) is an ML task of learning a function by mapping inputs to an output based on the example input-output pairs provided (Stuart J. Russell, 2010). SL algorithms classify objects that can be categorised and grouped up, which is suitable for estimating the relationship between different features in a data set.

Unsupervised learning (UL)
Unsupervised learning is when an ML algorithm is not provided with any pre-assigned labels or scores for training the dataset (Hinton & Sejnowski, 1999). As a result, UL algorithms are only provided with input and must discover the learning patterns to give an output. This means the ML algorithm learns from observations and finding structures within the data sets.

Reinforcement learning (RL)
Reinforcement learning uses intelligent agents to react to actions based on the environment to maximise the notion of cumulative reward (Hu, Niu, Carrasco, Lennox, & Arvin, 2020). A good analogy for RL is “training a dog”, where the dog is taught how to respond to specific gestures such as whistling or clapping. Whenever the dog responds correctly, the trainer gives it a reward. Applications of RL include natural language processing, playing games such as chess or self-driving vehicles.

4.1.2 Decision Tree Classifier (DTC)

A decision tree classifier organises the data set into a tree-like data structure that includes decision nodes as a predictive model. Given the simplicity, DTCs are ranked one of the most popular machine learning and data mining algorithms (Holzinger, 2015). DTC are simple to understand and interpret as the visual layout of a tree and observe how a decision was made by following the nodes and branches of the tree. A decision tree classifier uses a “white box model”, a given problem in the model is explainable by Boolean logic of true and false.

However, small variations in the provided dataset may result in a completely different decision tree being produced. Also, overfitting based on the data in a complex decision tree is a potential issue; a solution to this is pruning the leaves of the nodes.

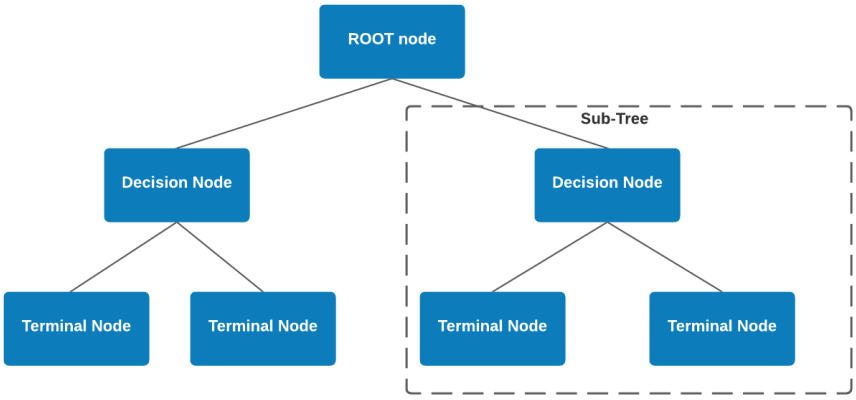

Figure 13. Example decision tree

4.1.3 Support Vector Machine (SVM) Classifier

SVMs are powerful supervised classification algorithms and classifies by finding the linear optimal separating hyperplane that splits all data points from different classes. Classes are identified by plotting points in n-dimension space where n are the number of features in the dataset (Gandhi, 2018). The linear optimal separating hyperplane is found using support vectors (training tuples) and margins (support vectors). Kernels allow the separation of non-linearly separable problems to be solved, and a linear classifier cannot solve most complex problems.

SVM is the most suitable for binary classification problems, with the ability to handle multi-class separation problems. Applications of SVM classification algorithms include text and hypertext categorisation, image classification, face detection, bioinformatics, and handwriting recognition (Gour, 2019). A major advantage of SVM classifiers over other classification algorithms is that they are less prone to noise unbalanced size of training data within each class (Talbot, 2010).

Figure 14 is an SVM example of a 2-dimensional separable problem, where the support vectors marked with grey squares describe the margins of larges separation between the two classes.

Figure 14. SVM example (Vapnik, 1995)

4.1.4 Multi-layer Perceptron Classifier (MLP)

MLPs are feedforward neural networks with one or more layers between an input and output layer. Each neuron in a layer is interconnected to every adjacent layer called the hidden layer used to process the input layer. MLPs use a supervised learning technique called backpropagation for training (Mohanty, 2019). MLPs mimic the human biological brains to solve computationally complex problems and could learn based on the representation of the training data set and relate it to the outputs based on mapping the variables.

Figure 15. Example of an MLP (Mohanty, 2019)

4.2 Prediction Model Performance

Each ML prediction model was evaluated based on their accuracy, precision, recall and f-score as performance metrics. For each instance, the decision making is categorised into one of these possible scenarios:

1. **True Positive (TP)** is where the sample is positive, and the classifier correctly labels the sample as positive.
2. **True Negative (TN)** is where the sample is negative, and the classifier correctly labels the sample as positive.
3. **False Positive (FP)** is where the sample is negative, but the classifier labels the sample as positive.
4. **False Negative (FN)** is where the sample is positive, but the classifier labels the samples as negative.

Table 5 presents the most representative measure of a classifier's performance.

Table 5. Measures used to assess each model's performance

Measure	Mathematical expression
Accuracy	$\frac{TP + TN}{TP + FN + FP + TN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F-score	$2 \times \frac{Precision \times Recall}{(Precision + Recall)}$

4.3 Configurations for ML Model

The parameters used for each ML model experiment are listed below in Table 6. To find the optimal parameters to use for each ML model, grid-searching was conducted using the Scikit learn implementation of GridSearchCV. The implementation can be found in Appendix C with the python source code. Grid-searching is the process of scanning the dataset to configure and calculate the best parameters to use for an ML model (Lutins, 2017). Grid searching is computationally expensive and can take a long time to run as it iterates through each parameter combination and stores a model for each combination.

Table 6. ML model's configuration

Classifiers used	Configurations
Decision Tree Classifier	**Evaluated grid-search parameters:** Criterion = 'gini' splitter = 'random' max_features = 'sqrt' **Default Scikit-learn parameters:** Criterion = 'gini', splitter = 'best', max_features='None', max_depth = None, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, random_state = None, max_leaf_nodes = None, min_impurity_decrease = 0.0,

	min_impurity_split = None, class_weight = None, ccp_alpha = 0.0
Support vector machine	**Evaluated grid-search parameters:** Kernel = 'rbf' C = '100' max_iter = '10' **Default Scikit-learn parameters:** Kernel = 'rbf', degree = '3', Gamma = 'auto', coef0 = 0.0, shrinking = True, probability = False, tol = 1e-3, cache_size = 200, class_weight = None, verbose = False, decision_function_shape = 'ovr', break_ties = False, random_state = None
Multi-layer perceptron	**Evaluated grid-search parameters:** Hidden_layer_size = (20,) activation = 'relu' solver = 'sgd' alpha = '0.05' learning_rate = 'constant' **Default Scikit-learn parameters:** Hidden_layer_size = '100', activation = 'relu', solver = 'adam', alpha = '0.001', batch_size = 'auto', learning_rate = 'constant', learning_rate_init = '0.001', power_t = '0.5', max_iter = 200, shuffle = True, random_state = None, tol = 1e-4, verbose = False, warm_start = False, momentum = 0.9, nesterous_momentum = True, early_stopping = False, validation_fration = 0.1, beta_1 = 0.9, beta_2 = 0.999, epsilon = 1e-8, n_iter_no_change = 10, max_fun = 15000

4.4 Model Valuation and Evaluation

After the fall's outcome classification process, it is important to evaluate and validate the accuracy of the classified outcomes for the severity of a fall. K-fold cross-validation is a data partitioning strategy applied to a dataset to help build more generalised ML models, with *k* being the number of sets the datasets are split into. The goal of cross-validation is to assist in developing more generalised ML models that perform well on unseen data.

After pre-processing the dataset, the final output was passed through 5-fold cross-validation, which was randomly split into independent 5 folds. The k-1 folds are used for the model training whole one-fold is used for performance evaluation. This process is carried out for 5 iterations, so a k number of performance estimates are obtained for each iteration and concluded with a mean k number of all the performance estimates. As the k-folds was 5, 20% of the test dataset is held back each time. Also, because the splitting process was done without replacement, each observation is used for both training and validation exactly once.

5-folds cross-validation was used over 10-fold cross-validation because the process is computationally intense and takes a long time. In the research case, one k-fold iteration was around 20 minutes, meaning for 5-folds would take roughly 1 hour 40 min and 10-folds would be double at over 3 hours.

4.5 Model Classification Results

This section presents the results for each ML model, which includes a classification report and the produced confusion matrix. Each class is represented by a number that correlates to the severity of a fall outcome; for example, 0 is equivalent to critical fall outcome whilst 3 relates to minor fall outcome in the confusion matrixes and classification reports.

A confusion matrix, also known as an error matrix, summarises the predicted results of a classification problem visualised in a table format. When observing a confusion matrix's, the diagonal intersections where the true label and predicted label intersect are the correctly predicted outcomes.

Evaluating ML model's performance using accuracy and recall are typical indicators for assessing the effectiveness of an ML Model. However, when it comes to identifying each particular class, in our case, the outcome for an elderly fall, other measures should be considered, such as F-score and Matthews Correlation Coefficient. F-score measures a model's accuracy on a data set and uses the mean of precision and recall metrics.

A Matthews Correlation Coefficient (MCC) score, which is a measure of the quality of binary (two-class) classification (Matthews, 1975). MCC is regarded as a balanced measure over accuracy and F1 score as it accounts for true and false positives and negatives. MCC returns a score between -1 and +1, where +1 denotes a perfect prediction, 0 is a random prediction, and -1 denotes predictions, and observations are in total disagreement.

4.5.1 Decision Tree Classifier Results

A decision tree visualisation is produced from the DTC model, and due to the sheer scale of the visualisation, it can be accessed in the link in Appendix D. Nevertheless, based on the output, we observe that the first decision node split is on *facility_n,* which are the locations of AnglicareSA facilities meaning some locations experience more falls than others.

Table 7. Classification Report for Decision Tree Classifier

	Precision	Recall	F1-score	Support
0 - Critical	0.00	0.00	0.00	25
1 - Insignificant	0.29	0.27	0.28	469
2 - Major	0.00	0.00	0.00	7
3 - Minor	0.72	0.76	0.74	1509
4 - Moderate	0.17	0.12	0.14	171
Accuracy			0.59	2181
Macro average	0.24	0.23	0.23	2181
Weighted average	0.57	0.59	0.58	2181

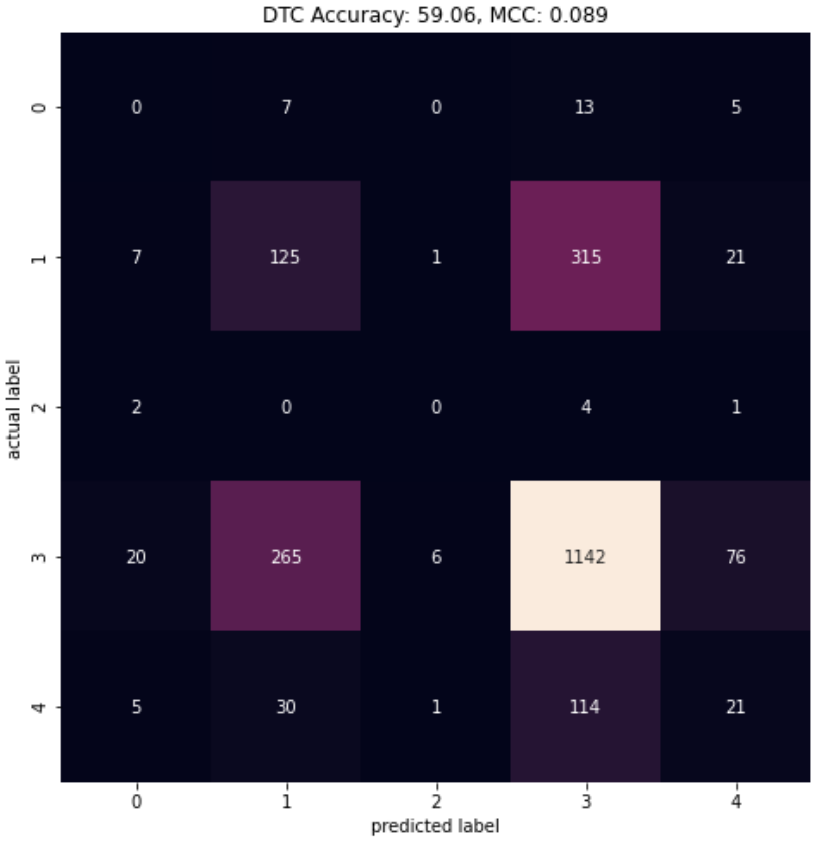

Figure 16. Confusion matrix for Decision Tree Classification Model

4.5.2 Multi-layer Perceptron Results

Table 8. Classification Report for Multi-layer Perceptron

	Precision	Recall	F1-score	Support
0 - Critical	0.00	0.00	0.00	25
1 - Insignificant	0.00	0.00	0.00	469
2 - Major	0.00	0.00	0.00	7
3 - Minor	0.69	1.00	0.82	1509
4 - Moderate	0.00	0.00	0.00	171
Accuracy			0.69	2181
Macro average	0.14	0.20	0.16	2181
Weighted average	0.48	0.69	0.57	2181

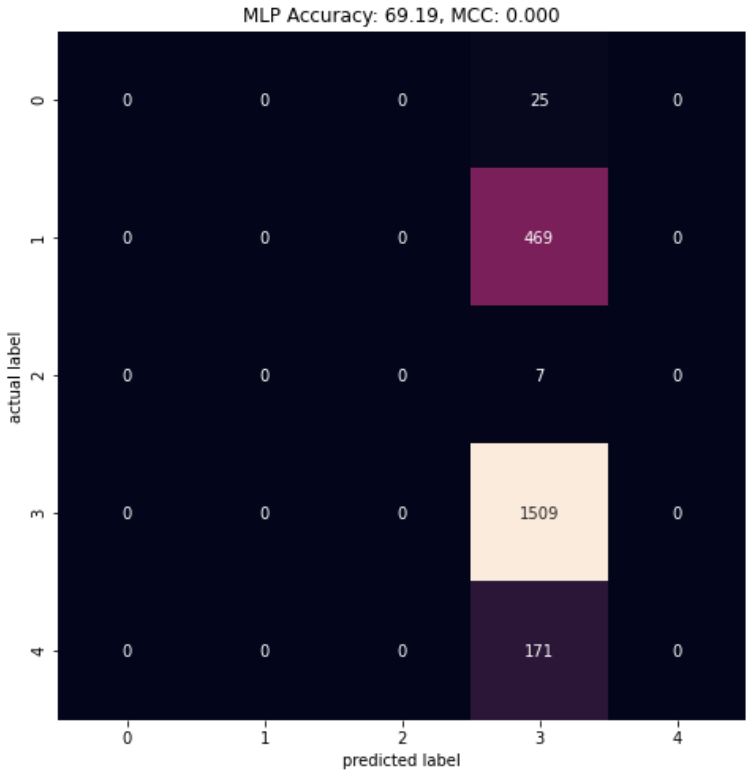

Figure 17. Confusion Matrix for Support-Vector Machine Model

4.5.3 Support-vector Machine Results

Table 9. Classification Report for Support-vector Machine

	Precision	Recall	F1-score	Support
0 - Critical	0.01	0.08	0.02	25
1 - Insignificant	0.21	0.28	0.24	469
2 - Major	0.00	0.00	0.00	7
3 - Minor	0.69	0.47	0.56	1509
4 - Moderate	0.05	0.05	0.05	171
Accuracy			0.39	2181
Macro average	0.19	0.18	0.17	2181
Weighted average	0.52	0.39	0.44	2181

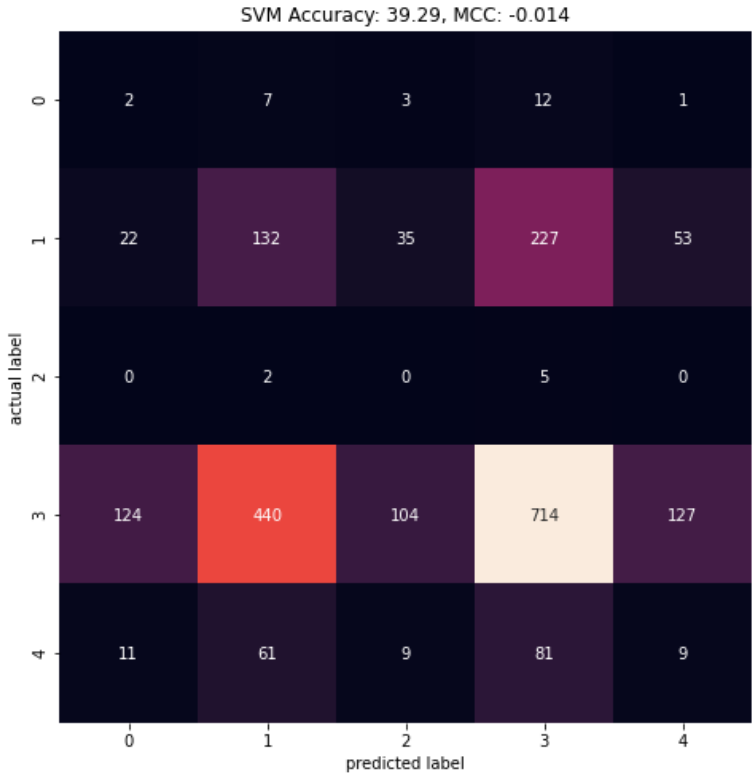

Figure 18. Confusion Matrix for Multi-layer Perceptron Model

4.2.1 Discussion of Results

The experiments have demonstrated that using DTC, SVM, and MLP models can somewhat predict the severity of a fall with an average accuracy of 60%, 69% and 39%, respectively. Though the results were moderate at best, it is nearly close to guessing highlighted by an MCC score below 0, which is unsuitable for building a fall prediction and prevention model for health applications that need high accuracy and MCC scores for adoption in clinical settings.

Table 10. Comparison of ML Model's

	DTC	MLP	SVM
Matthews Correlation Coefficient (MCC)	0.100	0.000	-0.014
Precision (weighted average)	0.58	0.48	0.52
Recall (weighted average)	0.60	0.69	0.39
F1-score (weighted average)	0.59	0.57	0.44
Accuracy	0.60	0.69	0.39

The reason for the intermediate results is attributed to the unbalanced nature of the final pre-processed dataset. This is demonstrated in Table 10 above, where nearly 70% of the 2,181 severity of a fall outcome came from the minor class, meaning that of all the falls that occurred, 70% of reported falls were minor. The unbalanced nature raises an issue that a classifier could classify all the fall severity outcomes as minor and score an accuracy of 70%, which seems to have occurred with the MLP, which scored an accuracy of 69%; however, the MCC is 0, indicating random predictions or guessing.

Another contributor to the poor results was the relatively small size of the dataset, with 2181 records spanning around 2 years' worth of fall incidents and clinical health records. Nevertheless, Section 5 suggests future work ideas that can be applied to similar research to help support further research in elderly falls prediction and prevention.

5. DISCUSSION

5.1 Limitations and Challenges

The research encountered a variety of challenges that were observed during the development and modelling process. Listed below are some of the limitations encountered.

- The dataset utilised for the study used data collected between 2019 and 2021, while other studies would use datasets with a longer time horizon. Generally, more data available over a long-time horizon leads to more reliable modelling and performance.
- The dataset used is specific to AnglicareSA aged care services context and operations. In a supervised model, most of the bias comes from the data, including where, how, and when the data was collected and by whom. Pre-processing of the dataset aims to fix missing values and derive values from the existing data. However, pre-processing ultimately changes the structure of the data and potentially introduces biases on what is suitable for the operator.
- Also, it is possible other fall-related features could assist in better explaining elderly falls, such as resident cognition, facilities design, facilities layout and more. Due to the lack of certain features, the prediction scores were reasonable at best.

5.2 Suggestions for Future Work

Studies in this area have great potential to predict and detect elderly falls, avoiding the negative impacts occurring from falls such as injuries, fractures, and death. Below are suggestions for future researchers to improve upon the study when applying ML techniques on similar projects.

1. The use of more extensive datasets from multiple age care providers on a country or even global scale, as elderly falls, is a global issue that requires datasets collected from different regions. Most open-source datasets such as the SisFalls (Sucerquia, 2017) use wearable or non-wearable devices to capture daily human activities, which ML models classify.
2. Improving data collection when using digital forms filled by clinicians. This can include mandatory fields, dropdowns, checkboxes on the forms to help collect essential information related to the study and reduce the number of errors such as missing information through a fall risk assessment (ACSQH, 2009)
3. Applying wearable devices as a data collection mechanism may include smartphones, smartwatches, or sensors, which are more effective at monitoring daily human activities in real-time and complementing a fall prediction and detection ML model. However,

considerations must be made on the cost of acquiring the devices and the resident's loss of privacy.

4. Combining different types of datasets to help predict and detect falls is called data diversity. An example is that instead of collecting elderly falls incidents only, which introduces a bias towards detecting falls only, information around what prevents falls could be a significant contribution. When collecting various incidences helps in identifying and discriminating falls when placed against other features.
5. Data visualisation ideas may include collecting the exact known locations of fall incidents and overlaying a heat map over a building blueprint or cross-section. Highlighting red zones where most fall incidents occur lead to preventative measures that can be taken to mitigate future falls. This can be achieved by using a clustering algorithm that classicises data points by grouping them into the closest related classes.

5.3 Conclusion

The research explored the application of ML techniques to predict the severity of a fall based on the fall incident reports and basic clinical health information datasets sourced from AnglicareSA. Three ML models were experimented with the fall incident reports and clinical health information datasets—the ML models included a decision tree classifier, support-vector machine, and multi-layer perceptron. The accuracy results of the DTC, SVM and MLP models were 60%, 69%, and 39%, respectively, which is moderate at best.

The results were meagre due to the relatively small size of the dataset after pre-processing, which was about 2181 records. Another factor for the poor results was the unbalanced nature of the dataset, where 70% of fall severity outcomes are minor, meaning a bias towards the minor class. Though the results were to be expected, the data sourced from AnglicareSA was initially not collected with applying ML models in mind and was the first attempt to examine existing datasets for the potential of developing an elderly fall prediction and prevention model.

Recommendations for future work include using a larger sample size or dataset to help with ML model classification. The more the samples, the more examples an ML model can work with. Another suggestion is to collect more relevant information contributing to falls when residents undergo clinical health assessments, including bone density, mass, and disease. A final recommendation would be investigating the use of wearable and non-wearable devices such as smartphones, smartwatches, and sensors as a data capture mechanism which from related studies have proven to be accurate in consistently capturing elderly movements and activities.

6. REFERENCES

ACSQH. (2009). *Preventing Falls and Harm From Falls in Older People.* Retrieved from https://www.safetyandquality.gov.au/sites/default/files/migrated/Guidelines-HOSP1.pdf

AIHW. (2019). Trends in hospitalised injury due to falls in older people 2007–08 to 2016–17.

Andrew J. Young, A. H. (2021). Using Machine Learning to Make Predictions in Patients Who Fall. *Journal of Surgical Research, 257*, 118-127. doi:https://doi.org/10.1016/j.jss.2020.07.047

Angela Sucerquia, J. D.-B. (2017). SisFall: A Fall and Movement Dataset. *Sensors, 17*. doi:https://doi.org/10.3390/s17010198

Australian Institute of Health and Welfare and Flinders University. (2019). *Trends in hospitalised injury due to.*

Aziz, O. M. (2017). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Med Biol Eng Comput, 55*, 45–55. doi:https://doi.org/10.1007/s11517-016-1504-y

Briana Moreland, R. K. (2020, July 10). *Trends in Nonfatal falls and fall-related injuries amoung adults aged 65 years*. Retrieved from https://www.cdc.gov/mmwr/volumes/69/wr/mm6927a5.htm?s_cid=mm6927a5_w

Casilari, E. &.-J. (2015). Automatic Fall Detection System Based on the Combined Use of a Smartphone and a Smartwatch. *PloS one, 10*(11). doi:https://doi.org/10.1371/journal.pone.0140929

Centers for Disease Control and Prevention. (2019). *National Vital Statistics Report.* Centers for Disease Control and Prevention (CDC).

Christina J. Charlesworth, E. S. (2015). Polypharmacy Among Adults Aged 65 Years and Older in the United States: 1988–2010. *Journals of Gerontology: Medical Sciences*, 989–995. doi:10.1093/gerona/glv013

Chu, I. F. (2019). *Data Cleaning.* ACM books.

David S. Lindberg, M. P. (2020). Identification of important factors in an inpatient fall risk prediction model to improve the quality of care using EHR and electronic administrative data: A machine-learning approach. *143*. doi:https://doi.org/10.1016/j.ijmedinf.2020.104272

Diana Yacchirema, J. S. (2019). Fall detection system for elderly people using IoT and ensemble machine learning algorithm. *Pers Ubiquit Comput*(23), 801–817. doi:https://doi.org/10.1007/s00779-018-01196-8

Faulkner, C. M. (2007). The history of falls and the association of the timed up and go test to falls and near-falls in older adults with hip osteoarthritis. *BMC Geriatrics*(17). doi:https://doi.org/10.1186/1471-2318-7-17

G. Plasqui, A. G. (2013). Daily physical activity assessment with accelerometers: new insights and validation studies. *Obesity Reviews, 14*(6), 451-462. doi:https://doi.org/10.1111/obr.12021

Gandhi, R. (2018, June 8). Support Vector Machine — Introduction to Machine Learning Algorithms. Retrieved from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Gour, R. (2019, March 8). 8 Unique Real-Life Applications of SVM. Retrieved from https://medium.com/@rinu.gour123/8-unique-real-life-applications-of-svm-8a96ca43313

Guyon I., E. A. (2006). An Introduction to Feature Extraction. *Feature Extraction. Studies in Fuzziness and Soft Computing, 207*. doi:https://doi.org/10.1007/978-3-540-35488-8_1

Hendrie D, M. T. (2012). *Incidence and costs of injury in Western Australia 2012.* Chronic Disease Prevention Directorate Department of Health WA. Government of Western Australia, Department of Health. Retrieved from https://ww2.health.wa.gov.au/~/media/Files/Corporate/Reports-and-publications/Cost-of-injury/Incidence-and-costs-of-injury-in-wa.pdf

Hinton, G., & Sejnowski, T. (1999). *Unsupervised Learning: Foundations of Neural Computation.* MIT Press.

Holzinger, A. (2015). Data Mining with Decision Trees: Theory and Applications. Online Information Review. doi:10.1108/OIR-04-2015-0121

Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2020, December). Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology, 69*(12), 14413-14423. doi:10.1109/TVT.2020.3034800

IBM. (2020). *what is Artificial Intekkigence (AI)*. Retrieved from https://www.ibm.com/au-en/cloud/learn/what-is-artificial-intelligence

IBM. (2020, July 15). *What is Machine Learning*. Retrieved from https://www.ibm.com/au-en/cloud/learn/machine-learning

J A Stevens, P. S. (2006). The costs of fatal and non-fatal falls among older adults. *Injury Prevention , 12*, 290–295. doi:10.1136/ip.2005.011015

Jaime Lynn Speiser, P. M. (2021, April 4). Machine Learning in Aging: An Example of Developing Prediction Models for Serious Fall Injury in Older Adults. *The Journals of Gerontology: Series A, 76*(4), 647–654. doi:https://doi.org/10.1093/gerona/glaa138

Lutins, E. (2017, September 6). Grid Searching in Machine Learning: Quick Explanation and Python Implementation. Retrieved from https://elutins.medium.com/grid-searching-in-machine-learning-quick-explanation-and-python-implementation-550552200596#:~:text=Grid%2Dsearching%20is%20the%20process,parameters%20for%20a%20given%20model.&text=Grid%2Dsearching%20can%20be%20applied,use%20for

Mark Sendak, M. G. (2019). Machine Learning in Health Care: A Critical Appraisal of Challenges and Opportunities. *The journal for Electronic Health Data and Methods*, 287. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6354017/

Mary B. King, M. a. (1995). Falls in community-dwelling older persons. *J Am Geriatr Soc*, 1146-1154.

Mary E. Tinetti, M. S. (1988). Risk Factors for Falls among Elderly Persons Living in the Community. *N Engl J Med*, 1701-1707.

Mastorakis, G. M. (2014). Fall detection system using Kinect's infrared sensor. *9*, pp. 635–646. J Real-Time Image Proc. doi:https://doi.org/10.1007/s11554-012-0246-9

Matthews, B. W. (1975, October 20). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure, 405*(2), 442–451. doi:https://doi.org/10.1016/0005-2795(75)90109-9

Microsoft. (2021). *Data Lake*. Retrieved from Azure Microsoft: https://azure.microsoft.com/en-gb/solutions/data-lake/?form=MY01SV&OCID=MY01SV

Miguel Ángel Álvarez de la Concepción, L. M.-A. (2017). Mobile activity recognition and fall detection system for elderly people using Ameva algorithm. *Pervasive and Mobile Computin, 34*, 3-13. doi:https://doi.org/10.1016/j.pmcj.2016.05.002

Mohanty, A. (2019, March 15). Multi layer Perceptron (MLP) Models on Real World Banking Data. Retrieved from https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f

Phelan E. A., M. J. (2015). Assessment and Management of Fall Risk in Primary Care Settings. *Medical Clinics of North America*, 281-293.

Pillai, S. B. (2020). Fall Detection for Elderly People using Machine Learning. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, (pp. 1-4). doi:10.1109/ICCCNT49239.2020.9225494

Pitts, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115-133. Retrieved from https://www.cs.cmu.edu/~./epxing/Class/10715/reading/McCulloch.and.Pitts.pdf

R. Alazrai, Y. M. (2015). A fall prediction methodology for elderly based on a depth camera. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (pp. 4990-4993). doi:10.1109/EMBC.2015.7319512

RLDatix. (2021). RiskMan. Retrieved 10 10, 2021, from https://rldatix.com/en-apac/riskman

S. Yang, W. Z. (2013). Fall-prediction algorithm using a neural network for safety enhancement of elderly. *2013 CACS International Automatic Control Conference (CACS)*, (pp. 245-249). doi:10.1109/CACS.2013.6734140

Samuel, A. L. (1959, July). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development, 3*(3), 210-229. doi:https://doi.org/10.1147%2Frd.33.0210

Santoyo-Ramón, J. C.-G. (2018). Analysis of a Smartphone-Based Architecture with Multiple Mobility Sensors for Fall Detection with Supervised Learning. *Sensors, 18*(4). doi:https://doi.org/10.3390/s18041155

Sollitto, M. (2021, September 16). *Agingcare*. Retrieved from Agingcare: https://www.agingcare.com/articles/falls-in-elderly-people-133953.htm

Stuart J. Russell, P. N. (2010). *Artificial Intelligence: A Modern Approach* (3 ed.). Prentice Hall. Retrieved from http://aima.cs.berkeley.edu/

Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python.* Academic Press. doi:https://doi.org/10.1016/B978-0-12-821379-7.00001-1

Sucerquia, A. L.-B. (2017). *SisFall: A Fall and Movement Dataset* (Vol. 17). Sensors (Basel, Switzerland). doi:https://doi.org/10.3390/s17010198

T. Mitchell, J. C. (1986). *Machine Learning: A Guide to Current Research.* Kluwer Academic Publishers.

Talbot, G. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in. *Journal of Machine Learning Research*(11), 2079-2107. Retrieved from https://jmlr.org/papers/volume11/cawley10a/cawley10a.pdf

Telstra. (2021). Retrieved from Teslstra Health: https://www.telstrahealth.com/

Unpingco, J. (2019). *Python for Probability, Statistics, and Machine* (2nd ed.). San Diego, CA, USA: Springer International Publishing. Retrieved from https://link.springer.com/content/pdf/10.1007%2F978-3-030-18545-9.pdf

Vapnik, C. c. (1995). Support-Vector Networks. *Machine Learning, 20*(3), 273-297. Retrieved from https://link.springer.com/content/pdf/10.1007/BF00994018.pdf

Weidong Min, H. C. (2018). Detection of Human Falls on Furniture Using Scene Analysis Based on Deep Learning and Activity Characteristics. *6*, pp. 9324-9335. IEEE. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8263164

World Health Organisation. (2021, April 26). *Falls*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/falls

Y. Wang, K. W. (2017, February 1). WiFall: Device-Free Fall Detection by Wireless Networks. *IEEE Transactions on Mobile Computing, 16*(2), 581-594. doi:https://doi.org/10.1109/ISACV.2018.8354084

7. APPENDICES

Appendix A: Glossary

Term	Definition
Accuracy	Is a ML evaluation measurement to determine which ML model is better at finding patterns and identifying relationships in the dataset's features.
Activity of Daily Living (ADL)	List of normal daily activities which could include walking, standing, rising from a sitting position, descending from a standing to a sitting position, picking up an object, ascending and descending stairs.
Ambulation	The act of moving about or walking
Area under the curve (AUC)	A ML model measuring tool to evaluate the ability of a classifier to differentiate between classes. The higher the AUC means the better the performance of a particular classifier differentiating between positive and negative classes.
Artificial Intelligence	Artificial Intelligence (AI) leverages the processing power or computers to mimic the decision making and problem-solving capabilities of human brains (IBM, what is Artificial Intekkigence (AI), 2020)
Data	A collection of observations.
Dataset	A is a structured collection of related sets of information.
Decision Node	When a node is split into further multiple nodes.
Decision Tree Classifier (DTC)	A supervised machine leaning technique is used for classification and regression problems. The nodes represent features in the dataset, whilst the branches represent the decision rules.
Deep Learning	Is a subset of artificial intelligence and machine learning that uses artificial neural networks and mimics human brain behaviour for learning patterns and correlations within data.
F1 Score	Shows the balance between the precision and the recall which measures a classifiers completeness.
iCareHealth	A Telstra Health product a clinical, medication, and workforce management software for residential aged care, disability, and community care providers. Renamed to Aged, Disability & Community Care (ADCC) and can be accessed on www.telstrahealth.com
K-nearest neighbour (K-NN)	Is a supervised machine leaning algorithm used for classification and regression problems. It classifies data by estimating how likely a data point is a member of a particular group nearest to it.
Machine Learning	Machine Learning (ML) is a branch of Artificial Intelligence and Computer Science (IBM, What is Machine Learning, 2020) which processes data using algorithms and data by imitating how humans would learn and gradually improve on the accuracy.
Models	An output of a learning algorithms training. Learning algorithms train models, which are used to make predictions.
Multi-Layer Perceptron (MLP)	A feedforward artificial neural network which generates a set of outputs based on the set of inputs.
Performance	An ML metric used to evaluate the performance of an ML model.

Precision	Is the number of True Positives divided by the number of true positives and False Positives. The ML model evaluation method measure a classifiers exactness and is also known as the positive predictive value.
Recall	Is the number of True Positives divided by the number of True Positives and False Negatives. The ML model evaluation method measures a classifiers completeness and is also known as the true positive rate.
RiskMan	An AnglicareSA internal incident management and reporting tool
Support Vector Machine (SVM)	A supervised machine leaning algorithm, used for classification, regression, and outlier's detection.
Train	Applying a learning algorithm to data using numerical approaches like gradient descent.

Appendix B: Dataset

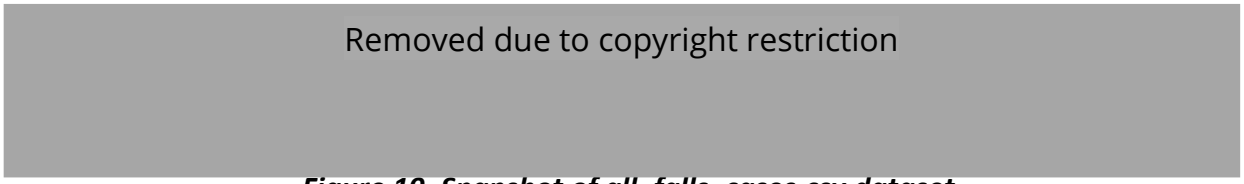

Figure 19. Snapshot of all_falls_cases.csv dataset

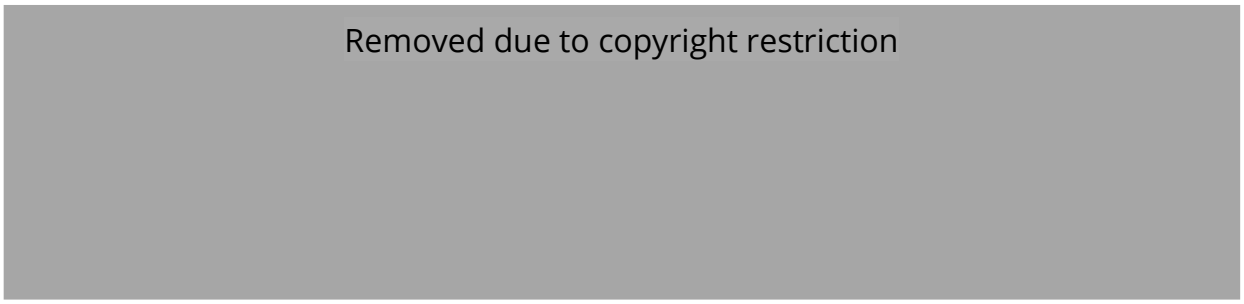

Figure 20. Snapshot of clinical_basic_info.csv dataset

Appendix C: Python Source Code

The full source code is available on a private GitHub repository at elderly falls prediction, the source code includes the datasets named *falls_incident_v2.xlxs* which contains the clinical health information and all the reported falls incidents.

Import all required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

Step 2. Data inspection

Explore and understand what data is available from the csv import

```
#Import data from csv
all_fall_cases = pd.read_excel('data/falls_incident_v2.xlsx', sheet_n
ame = 'all_fall_cases')
clinical_basic_info_load = pd.read_excel('data/falls_incident_v2.xlsx
', sheet_name = 'clinical_basic_info_load')

# Show first 10 rows in dataset
all_fall_cases.head(10)
# clinical_basic_info_load.head(10)
```

Step 2.1 Inspect all_fall_cases data set

```
# Describe all columns
all_fall_cases.describe(include='all')

# Inspect all_fall_cases dataset
all_fall_cases.iloc[:,1:].hist(bins=10,figsize=(15, 15))
plt.show()
```

Step 3.1 Clean clinical basic info dataset

- Renaming headings
- Measure deviation for age
- Group based on CareReceiverId
- Round numbers to 2 decimal points
- Remove rows with missing values

```
# 1. Rename some headings
rename_headings_clinical_info = clinical_basic_info_load.rename({'Res
piratory - rate':'RespiratoryRate'}, axis=1)

# 2. Remove unwanted columns
```

```
remove_columns_clinical_info = rename_headings_clinical_info.drop(['D
isplayID', 'ObserveredDate' ,'IncidentDate', 'BMI', 'Activity'], axis
=1)

# NOTE Deviation for majority of values was 0
# group_by_weight = rename_headings_clinical_info.groupby('CareReceiv
erId').agg({'Weight': ['mean', 'min', 'max']})
# group_by_height = rename_headings_clinical_info.groupby('CareReceiv
erId').agg({'Height': ['mean', 'min', 'max']})
# group_by_respiratory_rate = rename_headings_clinical_info.groupby('
CareReceiverId').agg({'Weight': ['mean', 'min', 'max']})

# 3. Group based on CareReceiverId
group_by_carereciverid_clinical_info = remove_columns_clinical_info.g
roupby('CareReceiverId').mean()
# 4.1 Add BMI derived from weight (kg) / {height (m)}^2
group_by_carereciverid_clinical_info['bmi'] = round(group_by_carereci
verid_clinical_info['Weight'] / (group_by_carereciverid_clinical_info
['Height']/100)**2,2)

# 5. Round numbers to 2 decimal points
group_by_carereciverid_clinical_info['RespiratoryRate'] = round(group
_by_carereciverid_clinical_info['RespiratoryRate'],2)
group_by_carereciverid_clinical_info['Weight'] = round(group_by_carer
eciverid_clinical_info['Weight'],2)

# 6. Remove rows with missing values
# print(group_by_carereciverid.isnull().sum()) 90 rows are removed
group_by_carereciverid_clinical_info.dropna(inplace=True)
print(group_by_carereciverid_clinical_info.shape)

clinical_basic_info_load = group_by_carereciverid_clinical_info
clinical_basic_info_load
```

Step 3.2 Clean All fall cases dataset

- Remove unwanted columns
- Remove all row missing values

```
# 1. Remove unwanted columns remove_columns_falls_cases =
all_fall_cases.drop(['DisplayId', 'BodyPartAffected', 'Facility_NumberOfBeds',
'CareReceiverId_1'], axis=1)

# 2. Remove all missing values rows
# print(remove_columns_falls_cases.isnull().sum())

# NOTE 1846 rows are removed
remove_columns_falls_cases.dropna(inplace=True)
# print(remove_columns_falls_cases.shape)
all_fall_cases = remove_columns_falls_cases all_fall_cases
```

Step 3.3 Merge all_fall_cases and clinical_basic_info datasets ¶

- Merge 2 datasets
- Remove rows with missing value

```
# Merge all_fall_cases and clinical_basic_info
falls_cases_and_clinical_data =
all_fall_cases.merge(clinical_basic_info_load, how='left',
left_on='CareReceiverId', right_on='CareReceiverId')

# falls_cases_and_clinical_data.head(10)
# falls_cases_and_clinical_data.describe(include='all')
# Clean merged data
# 1. Remove rows with missing value
# print(falls_cases_and_clinical_data.isnull().sum())
# NOTE 232 rows removed

falls_cases_and_clinical_data.dropna(inplace=True)
# falls_cases_and_clinical_data.isnull().sum()
# falls_cases_and_clinical_data.describe()
# NOTE 2187 rows left
falls_cases_and_clinical_data.head(3)
```

Step 4. Data Wrangling

- Transforming data values into numbers or categorical values
- Encoding labels to numbers
- Removing labels fields and leaving numbered fields

NOTE: ML algorithms only works on numbers

```
from sklearn.preprocessing import LabelEncoder

le_gender = LabelEncoder()
le_incidentLocation = LabelEncoder()
le_facility = LabelEncoder()
le_outcome = LabelEncoder()

# 1. Encoding labels to numbers
falls_cases_and_clinical_data['gender_n'] = le_gender.fit_transform(f
alls_cases_and_clinical_data['Gender'])
falls_cases_and_clinical_data['incidentLocation_n'] = le_gender.fit_t
ransform(falls_cases_and_clinical_data['IncidentLocation'])
falls_cases_and_clinical_data['facility_n'] = le_gender.fit_transform
(falls_cases_and_clinical_data['Facility'])
falls_cases_and_clinical_data['outcome_n'] = le_gender.fit_transform(
falls_cases_and_clinical_data['Outcome'])


# 2. Removing labels fields and leaving numbered fields
input_n = falls_cases_and_clinical_data.drop(['DateOfBirth', 'Admissi
onDate', 'IncidentDate', 'IncidentTime' ,'Gender','IncidentLocation',
'Facility', 'Outcome'],axis='columns') # Removes datetimes

# 3. Remove infinity values
```

```
input_n = input_n.replace([np.inf, -np.inf], np.nan).dropna(axis=0)
input_n.head()
```

Step 4.1 Inspect new dataset after pre-processing
NOTE: 2181 rows and 11 columns

```
# Check data quality after pre-processing
# input_n.info()

# NOTE 2181 rows and 11 columns
# input_n.shape

input_n.iloc[:,1:].hist(bins=10,figsize=(15, 15))
plt.show()
```

Step 5. Split the data into training and test datasets

- Allocating *20%* for testing
- Allocating *80%* for training

```
# Input set, Creates new dataset without genre (Question)
X = input_n.drop(columns=['outcome_n'])

# Output set, Creates new dataset with genre (Answer)
y = input_n['outcome_n']

# Allocating 20 % of the data for testing, and unpacking the tuple fo
r input & output testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0
.2)
```

Experiment 1: Decision Tree Classifier (DTC)

1.1 GridSearchCV for DTC

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, co
nfusion_matrix, plot_confusion_matrix

model = DecisionTreeClassifier()
# Train model so it learns patterns in the model using the input and
output set
model.fit(X_train, y_train)

male_prediction = ['18', '98.0', '6', '18.0', '44.8', '161.0', '17.28
3284', '0', '1', '5'] #Prediction Result 1
# male_prediction = ['82', '95.0', '2', '16.0', '79.5', '161.0', '30.
67', '0', '1', '5'] #Prediction Result3

dtc_prediction = model.predict([male_prediction])

# Calculate prediction accuracy
```

```
predictions = model.predict(X_test)
dtc_score = accuracy_score(y_test, predictions)
```

1.2 5-fold Cross Validation

```
from sklearn.model_selection import cross_val_score
from sklearn.utils import shuffle
from sklearn.model_selection import StratifiedKFold
from sklearn.base import clone

# shuffle the dataset every time we do the 5 CV to make it more rando
m
X_shuffle, y_shuffle = shuffle(X, y)

# create a pipeline classifier model based on the best parameters fro
m the GridSearchCV
dtc_model = clone(dtc_search.best_estimator_)

# stratification
skf = StratifiedKFold(n_splits=5,shuffle=True)

dtc_cross_val_scores = cross_val_score(dtc_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring='accuracy')

print('cross_val_score: {}'.format(dtc_cross_val_scores))
print("%0.2f accuracy with a standard deviation of %0.2f" % (dtc_cros
s_val_scores.mean(), dtc_cross_val_scores.std()))
```

```
# cross_validate returns more information and can use multiple scorer
s
from sklearn.model_selection import cross_validate

cross_validate_scores = cross_validate(dtc_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring=('accuracy','f1_macro'))
print('cross_validate: {}'.format(cross_validate_scores))
```

```
from sklearn.model_selection import cross_val_predict

y_pred = cross_val_predict(dtc_model, X_shuffle, y_shuffle, cv=skf, v
erbose=3)
print('cross_val_predict: {}'.format(y_pred))
```

1.3 DTC implementation with 5-fold Cross Validation

```
from sklearn.metrics import confusion_matrix from sklearn.metrics
import accuracy_score from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import classification_report import seaborn as
sns; print(classification_report(y_shuffle,y_pred)) acc =
accuracy_score(y_shuffle,y_pred) mcc =
matthews_corrcoef(y_shuffle,y_pred) print('Matthews Correlation
Coefficient: {}'.format(mcc)) mat = confusion_matrix(y_shuffle,
y_pred) fig, ax = plt.subplots(figsize = (10,8)) sns.heatmap(mat,
square=True, annot=True, fmt='d', cbar=False, ax=ax) plt.title('DTC
```

```
Accuracy: {:.2f}, MCC: {:.3f}'.format(acc*100,mcc))
plt.ylabel('actual label') plt.xlabel('predicted label')
```

1.4 DTC implementation train-test-split

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, co
nfusion_matrix, plot_confusion_matrix

male_prediction = ['18', '98.0', '6', '18.0', '44.8', '161.0', '17.28
3284', '0', '1', '5'] #Prediction Result 1
# male_prediction = ['82', '95.0', '2', '16.0', '79.5', '161.0', '30.
67', '0', '1', '5'] #Prediction Result3

dtc_prediction = dtc_search.predict([male_prediction])

# Calculate prediction accuracy
predictions = dtc_search.predict(X_test)
dtc_score = accuracy_score(y_test, predictions)

print(f"Prediction: {dtc_prediction[0]}")
print(f"Accuracy: {dtc_score*100}%")

print('Confusion matrix : \n')
plot_confusion_matrix(dtc_search, X_test, y_test)
plt.show()

print('Classification report : \n', classification_report(y_test,pred
ictions))
```

1.5 Visualising the Decision Tree Classifier

```
from sklearn import tree

# Creates a file for the decision tree
tree.export_graphviz(dtc_search.best_estimator_,
                     out_file='falls_prediction.dot',
#                    Columns for the data
                     feature_names=['CareReceiverId','Age','NumberOfIn
cidents','RespiratoryRate','Weight','Height','bmi','gender_n','incide
ntLocation_n','facility_n'],
                     label="all",
                     rounded=True,
                     filled=True)
```

Experiment 2: Multi-Layer Perceptron (MLP) Network

2.1 GridSearchCV for MLP

```
from sklearn.neural_network import MLPClassifier
from sklearn import svm
from sklearn.model_selection import GridSearchCV

# Create model and define grid of parameters to search over
param_grid  = {
```

```
    'hidden_layer_sizes': [(10,30,10),(20,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant','adaptive'],
}

mlp_model = MLPClassifier(max_iter=100)

# Set up the search
mlp_search = GridSearchCV(mlp_model, param_grid, cv=5, verbose=3, n_j
obs=-1)

# Perform the fitting/training over the param_grid
mlp_search.fit(X, y)

print('Best Parameters: {}'.format(mlp_search.best_params_))
print('Best Score (Mean Across Folds): {:.2f}'.format(mlp_search.best
_score_))
```

2.2 5-fold Cross Validation

```
from sklearn.model_selection import cross_val_score
from sklearn.utils import shuffle
from sklearn.model_selection import StratifiedKFold
from sklearn.base import clone

# shuffle the dataset every time we do the 5 CV to make it more rando
m
X_shuffle, y_shuffle = shuffle(X, y)

# create a pipeline classifier model based on the best parameters fro
m the GridSearchCV
mlp_model = clone(mlp_search.best_estimator_)

# stratification
skf = StratifiedKFold(n_splits=5,shuffle=True)

mlp_cross_val_scores = cross_val_score(mlp_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring='accuracy')

print('cross_val_score: {}'.format(mlp_cross_val_scores))
print("%0.2f accuracy with a standard deviation of %0.2f" % (mlp_cros
s_val_scores.mean(), mlp_cross_val_scores.std()))
```

```
# cross_validate returns more information and can use multiple scorer
s
from sklearn.model_selection import cross_validate

cross_validate_scores = cross_validate(mlp_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring=('accuracy','f1_macro'))
print('cross_validate: {}'.format(cross_validate_scores))
```

```
from sklearn.model_selection import cross_val_predict
```

```
y_pred = cross_val_predict(mlp_model, X_shuffle, y_shuffle, cv=skf, v
erbose=3)
print('cross_val_predict: {}'.format(y_pred))
```

2.3 Implement MLP using 5 CV

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import classification_report
import seaborn as sns;

print(classification_report(y_shuffle,y_pred))
acc = accuracy_score(y_shuffle,y_pred)
mcc = matthews_corrcoef(y_shuffle,y_pred)
print('Matthews Correlation Coefficient: {}'.format(mcc))

mat = confusion_matrix(y_shuffle, y_pred)
fig, ax = plt.subplots(figsize = (10,8))
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            ax=ax)
plt.title('MLP Accuracy: {:.2f}, MCC: {:.3f}'.format(acc*100,mcc))
plt.ylabel('actual label')
plt.xlabel('predicted label')
```

2.4 implementing MLP with train-test-split

```
mlp_prediction = mlp_search.predict(X_test)
mlp_score = mlp_search.score(X_test,y_test)

print(f"Prediction: {mlp_prediction[0]}")
print(f"Accuracy: {mlp_score*100}%",'\n')

mlp_confusion_matrix = confusion_matrix(y_test,mlp_prediction)
mlp_classification_report = classification_report(y_test,mlp_predicti
on)

print('Confusion matrix : \n')
plot_confusion_matrix(mlp_search, X_test, y_test)
plt.show()

print('Classification report : \n', mlp_classification_report )
```

Experiment 3: Support Vector Machine (SVM)

3.2 GridSearchCV for SVM

```
from sklearn import svm
from sklearn.model_selection import GridSearchCV

# Create model and define grid of parameters to search over
param_grid  = {
    'kernel':('linear', 'rbf'),
    'C':[1,10,100],
    'max_iter':[1,10,100]
}

svc_model = svm.SVC()

# Set up the search
svm_search = GridSearchCV(svc_model, param_grid, cv=5, verbose=3, n_j
obs=-1)

# Perform the fitting/training over the param_grid
svm_search.fit(X, y)

print('Best Parameters: {}'.format(svm_search.best_params_))
print('Best Score (Mean Across Folds): {:.2f}'.format(svm_search.best
_score_))
```

3.4 5 Cross-validation (CV)

```
from sklearn.model_selection import cross_val_score
from sklearn.utils import shuffle
from sklearn.model_selection import StratifiedKFold
from sklearn.base import clone

# shuffle the dataset every time we do the 5 CV to make it more rando
m
X_shuffle, y_shuffle = shuffle(X, y)

# create a pipeline classifier model based on the best parameters fro
m the GridSearchCV
svm_model = clone(svm_search.best_estimator_)

# stratification
skf = StratifiedKFold(n_splits=5,shuffle=True)

svm_cross_val_scores = cross_val_score(svm_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring='accuracy')

print('cross_val_score: {}'.format(svm_cross_val_scores))
print("%0.2f accuracy with a standard deviation of %0.2f" % (svm_cros
s_val_scores.mean(), svm_cross_val_scores.std()))
```

```
# cross_validate returns more information and can use multiple scorer
s
from sklearn.model_selection import cross_validate

cross_validate_scores = cross_validate(svm_model, X_shuffle, y_shuffl
e, cv=skf, verbose=3, scoring=('accuracy','f1_macro'))
print('cross_validate: {}'.format(cross_validate_scores))
```

```
from sklearn.model_selection import cross_val_predict

y_pred = cross_val_predict(svm_model, X_shuffle, y_shuffle, cv=skf, v
erbose=3)
print('cross_val_predict: {}'.format(y_pred))
```

3.5 Implement SVM using 5CV

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import matthews_corrcoef
from sklearn.metrics import classification_report
import seaborn as sns;

print(classification_report(y_shuffle,y_pred))
acc = accuracy_score(y_shuffle,y_pred)
mcc = matthews_corrcoef(y_shuffle,y_pred)
print('Matthews Correlation Coefficient: {}'.format(mcc))

mat = confusion_matrix(y_shuffle, y_pred)
fig, ax = plt.subplots(figsize = (10,8))
sns.heatmap(mat, square=True, annot=True, fmt='d', cbar=False,
            ax=ax)
plt.title('SVM Accuracy: {:.2f}, MCC: {:.3f}'.format(acc*100,mcc))
plt.ylabel('actual label')
plt.xlabel('predicted label')
```

3.6 Implement SVM using train-test-split

```
svm_search.fit(X_train, y_train)

famale_prediction = ['18', '98.0', '6', '18.0', '44.8', '161.0', '17.
283284', '0', '1', '5']
# famale_prediction = ['82', '95.0', '2', '16.0', '79.5', '161.0', '3
0.67', '0', '1', '5']

# Calculate prediction
svm_prediction = svm_search.predict([famale_prediction])

# Calculate accuracy
svm_score =  svm_search.score(X_test, y_test)

print(f"Prediction: {svm_prediction[0]}")
print(f"Accuracy: {svm_score*100}%")
```

```
print('Confusion matrix : \n')
plot_confusion_matrix(svm_search, X_test, y_test)
plt.show()
```

Appendix D: Decision Tree Classifier Visualisation

A decision tree was produced from the DTC ML model. However, due to the sheer scale of the decision tree visualisation, here is a link to a Microsoft oneDrive Folder with a *DTC-graphviz-visualisation.svg* file, which can be downloaded and viewed on any web browser.

Below is a snapshot of the *DTC-graphviz-visualisation.svg* used to classify the severity of a fall opened on Google Chrome.