Decoding Hospitalisation Predictors

A Data-Driven Analysis Using Logistic Regression in Multiple Myeloma Patients

María Victoria Friss de Kereki Tosar

September 2024

School of Mathematics,
Cardiff University

A dissertation submitted in partial fulfilment of the
requirements for MSc in Data Science and Analytics
by taught programme, supervised by Alexia Zoumpoulaki.

CANDIDATE'S ID NUMBER	22121504
CANDIDATE'S SURNAME	Please circle appropriate value Mr (Miss) Ms / Mrs / Rev / Dr / Other please specify Friss de Kereki Tosar
CANDIDATE'S FULL FORENAMES	María Victoria

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed V. Friss de Kereki (candidate) Date 08/09/2024

STATEMENT 1

This dissertation is being submitted in partial fulfillment of the requirements for the degree of MSc (insert MA, MSc, MBA, MScD, LLM etc, as appropriate)

Signed V. Friss de Kereki (candidate) Date 08/09/2024

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed V. Friss de Kereki (candidate) Date 08/09/2024

STATEMENT 3 – TO BE COMPLETED WHERE THE SECOND COPY OF THE DISSERTATION IS SUBMITTED IN AN APPROVED ELECTRONIC FORMAT

I confirm that the electronic copy is identical to the bound copy of the dissertation

Signed V. Friss de Kereki (candidate) Date 08/09/2024

STATEMENT 4

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed V. Friss de Kereki (candidate) Date 08/09/2024

STATEMENT 5 - BAR ON ACCESS APPROVED

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed V. Friss de Kereki (candidate) Date 08/09/2024

Acknowledgements

I want to express my deepest gratitude to the Foreign, Commonwealth and Development Office (FCDO), and the UK Embassy in Uruguay, for awarding me a Chevening Scholarship, which has made this journey possible. This life-changing experience, culminating in this project, would never have been realised without your generosity. As they say, "Chevening is more than just a year spent in the UK, it's a community for life,"[1] and I am honoured to be a part of it.

I am also immensely thankful to my family, especially my parents, Sylvia and Federico, and my friends in Uruguay, for supporting me in achieving my dreams. Your unwavering support, even from across the ocean, has been a pillar of strength throughout this journey.

A special thank you goes to my Cardiff friends, the Barbell Club, hockey teammates, MSc classmates, and fellow Cardiff Cheveners. You have made Cardiff my home and this year the best one ever. Your companionship and encouragement have been crucial in providing the support I needed.

I would also like to extend my gratitude to the United Kingdom as a whole, and Wales in particular, for being so welcoming. The warmth of its people and communities has made this experience truly memorable.

Once again, I want to thank everyone who has contributed to my academic journey. Your support and encouragement have meant the world to me, and I am excited to embrace the next chapter of my life with confidence and deep appreciation.

[1] Chevening, n.d.

Abstract

This project presents an analysis of hospitalisation risks among multiple myeloma patients using clinical data from a hospital. A logistic regression model is developed and refined to identify key hospitalisation predictors. The research explored various factors contributing to the hospitalisation of multiple myeloma patients, offering insights into risk factors and enhancing predictive modelling techniques in healthcare. The findings provide insights into hospitalisation risk factors and predictive modelling in healthcare settings.

Keywords: Multiple Myeloma, Hospitalisation Risk, Logistic Regression, Predictive Modelling, Disease Modelling

"*Medicine is a science of uncertainty and an art of probability*".

Sir William Osler (1849-1919)

Table of Contents

List of Tables

List of Figures

1. Introduction

This chapter describes the investigation's background, problem statement, study purpose, research questions, and the document's structure.

1.1 Background

Multiple myeloma (MM) is a blood cancer that originates in plasma cells. Significant challenges in patient management are faced due to its elaborate treatment regimens and complexities. One critical aspect of patient care recognising risk factors associated with potential hospitalisation. The development of a suitable risk model to predict hospital admissions can improve patient outcomes through individualised models and active intervention of possibly avoidable complications.

1.2 Statement of the Problem

Hospital admissions are a common and significant complication for patients undergoing treatment for MM, accruing costs without full quality of life and better results. Despite advancements in therapy, there is a lack of comprehensive models to predict which patients are at higher risk of hospitalisation. This hinders the ability to provide personalised care and proactive management, potentially leading to preventable admissions and suboptimal patient care. Thus, it becomes essential to develop a predictive risk model to assist caregivers in identifying patients at high risk of being hospitalised during MM treatment, guiding tailored interventions and improve overall patient outcomes.

1.3 Purpose of the Study

This investigation aims to create a risk model to estimate the chances of hospital admission for patients undergoing MM treatment. This model will compare data (from 01/10/2023 to 30/04/2024) of patients admitted to the University Hospital of Wales (UHW) during treatment, with data of patients of the same hospital, also on treatment, but not admitted. The objective is to identify and evaluate risk factors to determine which patients are at higher or lower risk of hospitalisation.

Potential risk factors to be considered in the model include:

- Patient Demographics: Age, sex, ethnicity, and economic class (based on postcode).

- Health Status: Comorbidities, weight, and the stage of MM.
- Medical History: Date of diagnosis, treatment received, and response to treatment.
- Disease-related factors: Degree of immuno-suppression, use of antibiotic and antiviral prophylaxis, and vaccination status.

By incorporating these factors, the risk model may provide insights into the variables most strongly associated with hospital admissions, leading to improved patient care by identifying individuals who may benefit from closer monitoring and tailored interventions during their treatment for MM.

1.4 Research Questions

The research questions that will be answered with this investigation are the following:

- Are patients more or less likely to be hospitalised as they advance in their treatments, considering factors such as treatment cycles and lines of therapy?[2]
- What demographic, clinical, and treatment factors contribute to the risk of hospital admission during MM treatment?[3]
- How effective is a predictive risk model in identifying MM patients at higher risk of hospitalisation?[4]

1.5 Structure

The document is structured as follows:

- Chapter 1, *Introduction*, offers context by outlining the background, identifying the problem, defining the study's objectives, and posing the research questions;
- Chapter 2, *Literature Review*, covers concepts such as the different types of MM, associated risk factors, and treatment options, as well as the application of statistical analysis in medical research, while also addressing ethical considerations.
- Chapter 3, *Methodology and Data*, details the nature of the study and the OSEMN methodology used, addresses ethical concerns, and outlines the employed tools.
- Chapter 4, *Obtain*, discusses the data acquisition process.

[2] This question is answered in Section 7.1 *Two-sample T-Test*.
[3] This question is answered in Section 7.2.1 *Hospital Admission Risk Factors Identification*.
[4] This question is answered in Section 7.2.2 *Hospitalisation Prediction*.

- Chapter 5, *Scrub*, focuses on the data cleaning process, explaining how missing values, outliers, and inconsistencies were handled and how the data were prepared for analysis, describing the needed transformations and preprocessing steps.
- Chapter 6, *Explore*, delves into the exploratory data analysis to uncover patterns, trends, and relationships within the data, through descriptive statistics, visualisations, and correlation analysis.
- Chapter 7, *Model*, develops the paired t-test analyses and logistic regression model used to identify and predict key hospitalisation risk factors.
- Chapter 8, *Interpret*, focuses on analysing and interpreting the findings from the developed models, offering insights into their implications.
- Chapter 9, *Conclusions and Future Work*, summarises the study's main findings, discusses their implications, and suggests areas for further research.

The project concludes with a bibliography, a glossary, and appendices that include code snippets related to various aspects, such as data cleaning, statistical analysis, and model implementation.

2. Literature Review

This chapter provides an overview of the current knowledge and research related to MM. It begins with a detailed profile of the disease, its pathophysiology, epidemiology, clinical manifestations, and classification. It looks into the key risk factors of the disease and analyses current lines of treatment. Additionally, it steps into the application of statistics in medical research, especially concerning MM studies, and the inherent ethical considerations.

2.1 Multiple Myeloma

MM is a haematological cancer characterised by the malignant proliferation of plasma cells in the bone marrow. Plasma cells produce antibodies, but malignant plasma cells produce abnormal monoclonal proteins. Such a change disrupts normal blood cell production and leads to systemic complications (NHS, 2021).

The pathophysiology of MM involves the uncontrolled growth of malignant plasma cells within the bone marrow, leading to excessive amounts of monoclonal proteins. This proliferation displaces normal blood cells and compromises bone integrity, often resulting in bone pain and fractures. Excessive monoclonal protein accumulation can also lead to injury of the kidneys and other organs (NHS, 2021).

Epidemiologically, MM is relatively rare, accounting for approximately 2% of all cancers and 15% of blood ones. The disease primarily affects older adults, with a median age at diagnosis around 70 years (MyelomaUK, n.d.). It exhibits a higher incidence among African Americans compared to other racial groups, highlighting demographic disparities in disease prevalence and outcomes (Bhutani, et al., 2023).

Clinical manifestations of MM are quite varied, with its most common being bone pain, fatigue, recurrent infections, hypercalcemia, renal impairment, and neurological deficits. Diagnosis typically involves blood and urine tests, bone marrow biopsies, and imaging modalities such as X-rays, MRI, or CT scans (Cancer Research UK, 2023; Blood cancer UK, n.d.).

2.1.1 Types of Multiple Myeloma

MM manifests in several forms, categorised by the type of abnormal immunoglobulin (protein) produced by the myeloma cell. The most common type is IgG myeloma, followed by IgA myeloma, determined by the heavy chain of the immunoglobulin involved. Less

common types include IgD, IgE, and IgM myelomas (The International Myeloma Foundation, 2021).

Additionally, there are unique forms of MM like light chain myeloma, where only the light chain components of the immunoglobulins are produced, and non-secretory myeloma, where the malignant cells produce little or no detectable immunoglobulin, making it harder to detect. Lastly, IgM myeloma is a very rare subtype and resembles lymphoma, a cancer of the lymph nodes, rather than myeloma, which primarily affects the bone marrow (Cancer Research UK, 2023).[5]

2.1.2 Risk Factors

This literature review synthesises current research on the diverse risk factors associated with MM.

- Genetic predisposition (Sergentanis, et al., 2015; Benjamin, Reddy, & Brawley, 2003)
- Lifestyle choices, including obesity (Becker, N., 2011; Brown, et al., 2001)
- Infectious agents, such as human herpesvirus-8 (HHV-8), although their exact role requires further investigation (Mackenzie, et al., 1997)
- Age, with most cases diagnosed in individuals over 65 (The American Cancer Society medical and editorial content team, 2024; Alexander, et al., 2007)
- Ethnicity; for instance African Americans are found to have a higher risk compared to Caucasians (Benjamin, Reddy, & Brawley, 2003; Ries, et al., 2006)

Understanding these multifaceted risk factors is crucial for developing targeted prevention strategies and personalised treatment approaches. Continued research into the interactions between genetics, environmental exposures, lifestyle choices, and infectious agents may help reduce MM incidence and improve patient outcomes.

2.1.3 Treatment

MM treatment can effectively manage the disease, alleviate symptoms and complications, and extend life, though it remains incurable (MyelomaUK, n.d.). Its treatment has evolved significantly over the years, improving patient outcomes (Monteith, Sandhu, & Lee, 2023).

[5] This section is particularly relevant to the preprocessing methods used in Section 5.2.7 *PP Type*.

The selection of a particular therapy for MM is influenced by factors such as the patient's age, overall health, existing medical conditions, and whether they can undergo stem cell transplantation. Additionally, another key factor is the risk classification of patients into standard or high-risk categories (Rajkumar & Kumar, 2020).

The NICE[6] guideline NG35[7] outlines several treatment options (NICE, 2018):

- Chemotherapy: Utilises drugs to kill or slow the growth of cancer cells.
- Targeted Cancer Drugs: Specifically target cancer cells with minimal damage to normal cells.
- Steroids: Reduce inflammation and directly kill myeloma cells.
- Stem Cell Transplantation: Involves replacing damaged bone marrow with healthy stem cells.
- Radiotherapy: Uses high-energy radiation to target and kill myeloma cells.
- Supportive Treatments: Include medications and therapies to manage symptoms and complications such as bone pain, anaemia, and infections. These treatments help improve the quality of life but do not treat MM itself.

2.2 Statistical Analysis in Medical Research

Statistical analysis plays a crucial role in clinical and experimental medical research by enabling researchers to make sense of complex data and to draw meaningful conclusions. It involves applying methods to design studies, analyse data, and interpret results, to ensure scientific validity, reliability, and reproducibility of research findings (Panos & Boeckler, 2023).

2.2.1 Key Statistical Techniques

Descriptive statistics, such as means, medians, and frequencies, summarise and describe the main features of a dataset. Inferential statistics, including t-tests, correlation, and regression, allow researchers to make generalisations from sample data to a larger population and test hypotheses about relationships between variables (Guetterman, 2019).

[6] "The National Institute for Health and Care Excellence (NICE) provides national guidance and advice to improve health and social care. NICE is an executive non-departmental public body, sponsored by the Department of Health and Social Care" (GOV.UK, n.d.).
[7] Myeloma: diagnosis and management (NICE, 2018)

Logistic regression has become a fundamental statistical tool in medical research, particularly over the past two decades. Widely employed to predict binary outcomes, such as hospitalisation or the presence of a disease, it evaluates the effects of various predictors and adjusts for confounding factors (Boateng & Abaye, 2019).

When evaluating logistic regression models, key metrics include *Accuracy*, *Precision*, *Recall*, and *F1 Score* (Sokolova & Lapalme, 2009):

- Accuracy: Measures the proportion of correct predictions out of the total predictions.
- Precision: Indicates the proportion of true positives out of all predicted positives.
- Recall (Sensitivity): Calculates the proportion of true positives relative to all actual positives.
- F1 Score: Balances precision and recall by calculating their harmonic mean.

In addition to these metrics, a *confusion matrix* is employed to evaluate the model's performance. This is a valuable tool for visualising and summarising predictions, displaying the distribution of true positives, true negatives, false positives, and false negatives (Singh, Singh, Singh, & Singh, 2021).

A *t-test* evaluates whether there is a significant difference between the means of two groups. It is typically applied when experimental subjects are split into two groups, where one receives treatment A and the other receives treatment B (Kim, 2015). In clinical research, comparing means between treatment groups often requires selecting the appropriate t-test based on the nature of the samples.

The *two-sample t-test* is used for comparing independent groups, while the *paired t-test* is suited for scenarios where observations are related, such as "before and after" studies or comparisons between matched pairs like twins (Xu, et al., 2017). A *two-tailed t-test* checks for differences in both directions, splitting the significance level between the two tails, and is used to test whether a value is either greater or less than a given value. In contrast, a *one-tailed t-test* focuses on one direction, allocating the entire significance level to a single tail. This approach provides more power to detect an effect in the specified direction but should only be used when the effect in the opposite direction is not of concern (UCLA: Statistical Consulting Group, 2024).

The traditional *Student's t-test* assumes equal variances between two groups, though it remains reasonably robust when standard deviations are similar. For improved accuracy,

especially when variances differ, the *Welch t-test* is recommended (and is used in this project). This alternative test accommodates unequal variances and provides comparable statistical power. Best practice suggests directly applying the Welch t-test rather than testing for variance differences (West, 2021).

2.2.2 Applications in Medical Research

Statistical methods are used in various aspects of medical research, including:

- Identifying Risk Factors: Statistical analysis helps identify risk factors associated with diseases and adverse outcomes. For example, a study used logistic regression to determine factors predicting hospitalisation in people with diabetes, analysing data from earlier examinations to identify significant predictors (Moss, Klein, & Klein, 1999).
- Evaluating Treatment Efficacy: Clinical trials rely heavily on statistical techniques to compare treatment groups and assess the efficacy and safety of new therapies, ensuring that observed differences are statistically significant and not due to random variation (Armitage, Berry, & Matthews, 2001).
- Predictive Modelling: Statistical models predict patient outcomes based on historical data. For example, predictive modelling can be used to plan healthcare services by predicting disease prevalence and future service demand (Su, Jaki, Hickey, Buchan, & Sperrin, 2016).

2.2.3 Statistical Analysis in Multiple Myeloma Research

Statistical analysis is critical in advancing the understanding of MM, particularly in refining treatment strategies and predicting patient outcomes. Recent research highlights the importance of robust statistical methods in evaluating the effectiveness of novel therapies and understanding disease progression. For instance, a study emphasises how statistical models are utilised to analyse clinical trial data, allowing for precise assessment of treatment efficacy and safety (Palumbo, et al., 2015).

Further developments in statistical methodologies have become key in handling the complex data associated with MM. Considering the disease's complex pathogenesis, the integration of serology, histology, radiology, and genetic data results in extensive high-dimensional datasets that exceed the capabilities of traditional analytical methods. Advanced computational

techniques, particularly Artificial Intelligence (AI) tools such as machine learning and deep learning, are increasingly relevant. These tools enhance data processing and analysis, improving diagnosis, prognosis, and treatment response evaluation in MM. Leveraging these methods allows researchers to refine analyses and contribute to more effective treatment protocols (Allegra, et al., 2022).

2.2.4 Ethical Considerations

Ethical considerations are paramount in medical research, to safeguard the research participants' dignity, rights, and well-being (World Health Organization, n.d.). Various organisations and authorities have developed guidelines to ensure integrity, adherence, and ethical practices in research activities (Yip, Han, & Sng, 2015).

Ethical concerns about AI in healthcare primarily revolve around data privacy and potential misuse. As technology evolves, maintaining patient anonymity is increasingly difficult. Early issues in AI precision medicine highlight risks related to data abuse and re-identification. Although data sharing benefits AI advancement, it can jeopardise patient confidentiality, by failing to protect sensitive health information (Hill, Data Privacy and Protection in AI for Precision Medicine, 2023).

Ethical issues also arise from biased datasets, which can distort outcomes and affect certain patient groups negatively. Statistical methods rely on accurate data to predict responses and outcomes, so data must reflect the diverse patient population. Researchers and clinicians must address biases in medical records to avoid improper diagnosis and treatment. Evidence shows that marginalised groups can be disproportionately impacted, highlighting concerns about racial and sex-specific disparities in healthcare (Hill, Bias in AI for precision medicine, 2023).

Applying statistical models ethically involves ensuring that analyses are transparent, reproducible, and replicable, as these principles are crucial for scientific integrity and advancing knowledge (Bakken, 2019).

3. Methodology and Data

This chapter outlines and justifies the methodology and data employed in the research process to guarantee the validity and reliability of the research findings. It describes the investigation type and a rationale for the chosen methodology, presents the study's ethical considerations, and introduces the used tools.

3.1 Type of Investigation

This investigation is an empirical research project, meaning it relies on the collection and analysis of real-world data to address specific research questions.

> *The term empiricism refers to making observations to obtain knowledge. (...) The term empirical research refers to making planned observations. (...) First, we need to plan what to observe. (...) Second, we need to plan whom to observe. (...) Third, we need to plan how to observe. (...) Next, we need to plan when to observe. (...) Finally, we should plan how to analyze the data and interpret them.*
>
> Patten, 2017

Additionally, this study employs a quantitative research design, focusing on numerical data to identify patterns and relationships.

> *QUANTITATIVE RESEARCH encompasses a range of methods concerned with the systematic investigation of social phenomena, using statistical or numerical data. Therefore, quantitative research involves measurement and assumes that the phenomenon under study can be measured. Quantitative research sets out to gather data using measurement, to analyse this data for trends and relationships and to verify the measurements made.*
>
> Watson, 2015

The quantitative method provides a structured framework for analysing data, ensuring that results are based on rigorous numerical analysis.

3.2 Methodology Used: OSEMN

This study follows the OSEMN framework, a structured methodology for data analysis (Mason & Wiggins, A Taxonomy of Data Science, 2010). This framework is widely used in Data Science projects (Brandt, 2016).

> *I'm sure this seems completely obvious to everybody in this room, that you get some data, you clean it up, you look at it, you interpret it, model it, and then you visualise it, or communicate it in some way. That's what it was. But in 2010, that was not obvious, and we wrote it down (...) and said, "We're going to say, 'This is the process. This is what you do when you are data science-ing.'" And it is really funny to talk about it here today, because now you look at it and you think, "Oh, this is so obvious. Like, of course." But it wasn't obvious in 2010.*
>
> Hilary Mason, 2022

This framework offers a systematic approach to data analysis, ensuring comprehensive coverage and flexibility to tailor it to specific project requirements (Lau, 2019). OSEMN is not inherently better or worse than other methods; however, its widespread use and proven efficacy in Data Science projects establish it as a dependable method. Following OSEMN enables researchers to gain a thorough understanding of their research questions by analysing all relevant aspects of the data (Lao, 2017).

The OSEMN methodology includes five steps: *Obtain*, *Scrub*, *Explore*, *Model*, and *iNterpret* (Mason & Wiggins, A Taxonomy of Data Science, 2010):

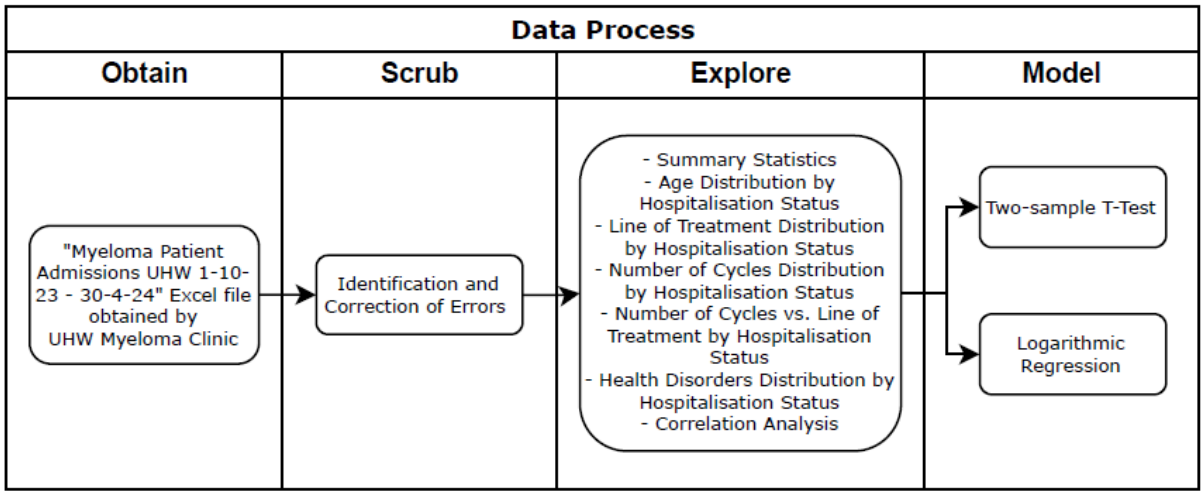
1) *Obtain*: Data are gathered.
2) *Scrub*: Data undergo cleaning and preprocessing, which involves addressing missing values, managing outliers, and transforming the data into an appropriate format for analysis.
3) *Explore*: Data are visualised and analysed to extract insights and discern patterns or relationships.
4) *Model*: Statistical or machine learning models are constructed to predict outcomes or categorise data.
5) *iNterpret*: The findings of the analysis are effectively communicated to stakeholders, typically through data visualisation or storytelling methods.

Figure 1 illustrates these steps.

Figure 1 *Data Science OSEMN methodology*

This sequence is designed to be a flexible framework adaptable to various Data Science projects, including the one outlined in this document. Figure 2 depicts the first four stages of OSEMN as applied in this project.

Figure 2 *Applied OSEMN methodology*



The *iNterpret* stage, which is broader and involves extracting insights from data rather than just processing it, is not shown due to its more comprehensive nature.

3.3 Ethical Considerations

Ethical considerations are fundamental in ensuring the responsible conduct of research. For this study, an Application for Ethical Review was filled by the author and the supervisors who collaborated on the research. This application was submitted to the Cardiff University School of Mathematics Ethics Board, which granted Ethical Approval.

The data used in this study were anonymised before being provided for research purposes, making it safe for use in research, participants' confidentiality maintained, and possible

privacy threats minimised. These measures reflect a strong commitment to maintaining high ethical standards and responsible data management through the entire research process.

3.4 Tools and Packages Used

This chapter introduces the tools and packages used in this investigation. All aspects of the project, including data collection and analysis, were carried out using the *Python* programming language.[8]

> *Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. (...) Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.*
>
> python.org, 2024

The following packages were used:[9]

- *geopy*[10] provides geographical operations, such as converting postcodes to geographical coordinates and calculating distances between coordinates.
- *matplotlib*[11] enables data visualisation through charts and plots, such as stacked bar charts and histograms.
- *numpy*[12] facilitates numerical operations and array computations.
- *pandas*[13] provides data manipulation and analysis capabilities, handling and analysing data structures such as DataFrames.[14]
- *re*[15] offers regular expressions for string operations, such as pattern matching.
- *seaborn*[16] enhances data visualisation with advanced plots, such as heatmaps for correlation matrices.
- *sklearn*[17] encompasses tools for machine learning and model evaluation tasks, such as:

[8] https://www.python.org/
[9] See Appendix B – *Python Setup*.
[10] https://geopy.readthedocs.io/en/stable/
[11] https://matplotlib.org/stable/index.html
[12] https://numpy.org/doc/
[13] https://pandas.pydata.org/docs/
[14] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
[15] https://docs.python.org/3/library/re.html
[16] https://seaborn.pydata.org/
[17] https://scikit-learn.org/stable/

- *accuracy_score*, *classification_report*, *confusion_matrix* for general model evaluation metrics.
- *make_scorer*, *f1_score* for F1 score calculation.
- *precision_score*, *recall_score* for precision and recall metrics.
- *train_test_split* to split data into training and testing sets.
- *GridSearchCV* for hyperparameter optimisation.
- *LogisticRegression* for logistic regression modelling.

- *warnings*[18] is used to manage and suppress non-critical warnings, ensuring that unnecessary alerts do not disrupt the execution of the code.

[18] https://docs.python.org/3/library/warnings.html

4. Obtain

The necessary data for analysis are gathered in OSEMN's *Obtain* phase. For this project, the (anonymised) dataset is sourced from the Myeloma Clinic at the University Hospital of Wales (UHW), covering the period from October 1st, 2023, to April 30th, 2024, and provided in the "`Myeloma Patient Admissions UHW 1-10-23 - 30-4-24.version 20-8-24.xlsx`" Excel file. The researcher did not directly participate in the data collection process due to the project's scope. Instead, healthcare professionals at UHW oversaw the data collection, adhering to ethical guidelines and data protection regulations.

The dataset was organised into two distinct tables:

A. Hospitalised Patients:
   - Patient Number: Unique patient identifier (1-36).
   - Date of Birth (DOB): Patient's birth date.
   - Gender: Patient's gender ("Male" or "Female").
   - Postcode: Area code where the patient resides.
   - Ethnicity: Patient's ethnic background (e.g., "African" or "Caucasian").
   - Current Treatment Regime: Ongoing treatment the patient is receiving, from a list of twenty options (e.g., "Len/Dex" for Lenalidomide/Dexamethasone).
   - Date of Admission: Date when the patient was admitted.
   - Day of Week: Day of the week the admission took place.
   - Reason for Admission: Reason for the patient's admission, from a list with eighteen options (e.g., "Temperature/Infection" or "Stroke").
   - Date of Discharge: Date when the patient was discharged.
   - Length of Stay (days): Days the patient stayed in the hospital.
   - Outcome: Patient's status after discharge (e.g., "Home" or "Palliative Care").
   - Date of Death (DOD): Date of death if applicable.
   - Survival Post Discharge: Days the patient survived after discharge, applicable only if the patient died.
   - Treatment Cycle Number: Cycle number of the current treatment.
   - Line of Treatment: Current line of treatment the patient is undergoing.

- Pre Admission Immunoglobulins: Immunoglobulin levels before admission, including IgG (g/l), IgA (g/l), IgM (g/l), KLC (mg/l), LLC (mg/l), PP Type, and PP Level.
- Post Admission/Post Discharge Immunoglobulins: Immunoglobulin levels after admission or post-discharge, including the same parameters as in Pre Admission Immunoglobulins.
- Baseline Neuts: Baseline neutrophil count.
- Antibiotic/Antiviral Prophylaxis Given: All antibiotic or antiviral prophylaxis administered.
- Comorbidities: Co-existing medical conditions.

B. Not Hospitalised Patients:
   - Number: Unique patient identifier (1-95).
   - Date of Birth (DOB): Patient's date of birth.
   - Number: Another unique, alternative patient identifier.
   - Gender: Patient's gender ("M" or "F").
   - Ethnicity: Patient's ethnic background (e.g., "Bangladeshi", or "Caucasian").
   - Postcode: Area code where the patient resides.
   - Diagnosis: Always "Myeloma".
   - Isotype: Isotype associated with the patient's diagnosis (e.g., "IgG Lambda", or "IgG Kappa")
   - Treatment: Treatment regimen the patient is undergoing (e.g., "DRD" or "IRD").
   - Line of Therapy: Current line of therapy the patient is undergoing.
   - Number of Cycles (30th April 24): Number of treatment cycles the patient has undergone as of April 30, 2024.
   - Immunoglobulins: PP, IgG, IgA, IgM, Neuts, KLC, LLC, and Ratio (KLC/LLC).
   - Comorbidities: Co-existing medical conditions.
   - Alive: Always "Y" for Yes.

Before handing the data for research, hospital personnel applied anonymisation procedures to it, removing all personally identifiable information while maintaining data integrity.

Once the anonymised datasets were received, they were processed and unified using Python, creating a cohesive platform to effectively address this study's research questions and objectives.[19]

[19] See Appendix C – *Dataset Importation*.

5. Scrub

This stage focused on preparing the data by conducting a thorough quality review and correcting errors to ensure the dataset was clean and consistent for analysis. Due to manual entry by hospital staff, the datasets contained significant inconsistencies, both internally and between them, requiring extensive cleaning to correct.

5.1 Identification and Correction of Errors

A critical aspect of data scrubbing involved identifying and correcting various types of errors, such as:

- Typographical Errors: Due to manual data entry, these errors included misspellings, incorrect data formats, and other similar issues. For example, some postcodes were entered without proper spacing: “CF33NW” instead of “CF3 3NW”.
- Inconsistent Data Entries: Different personnel used varying formats for the same information. For example, dates of birth were recorded in multiple formats: “23-Sep-1946” or “09/12/1947.”
- Missing Values: Some fields were left blank, necessitating imputation or other methods to address the gaps.
- Column Name Discrepancies: The datasets used different column names for identical information, requiring a renaming process for consistency before merging. For example, the Admitted Patients dataset used “PP Type”, while the Non-Admitted Patients dataset used “Isotype.”

5.2 Challenges and Solutions

One of the main challenges was managing the volume of inconsistencies and errors due to manual data entry. No single automated solution could resolve all issues, but a set of targeted scripts effectively handled most errors. These flexible scripts were designed to address a wide range of issues without manual intervention.[20]

5.2.1 Column Names and Dataset Unification

To address differing column names, a mapping was created to rename columns consistently across the two datasets, so when the datasets were joined, the merged dataset was coherent

[20] See Appendix D – *Dataset Concatenation* and Appendix E – *Data Cleaning*.

and usable. The creation of a *Hospitalised* column provided a clear distinction between the two groups of patients, facilitating accurate analysis and modelling.

Once the column names were standardised, the datasets were unified. This integration allowed for a comprehensive view of all patient records, enhancing the overall consistency and usability of the data for subsequent analysis.

5.2.2 Missing Values for Pre- and Post-Admission Data

When pre- and post-admission values were available, missing pre-admission data were replaced with post-admission values if present. Any remaining gaps were filled using values from other records of the same patient, ensuring consistency within each individual's dataset. This method, applied to columns *PP Type*, *PP Level*, *IgG (g/l)*, *IgA (g/l)*, *IgM (g/l)*, *LLC (mg/l)*, and *KLC (mg/l)*, aimed to make the dataset as complete and accurate as possible. By using data from the same patient, this approach reduced data loss and preserved the dataset's integrity and usability for analysis.

5.2.3 Ratio Calculation (KLC/LLC)

The *Ratio (KLC/LLC)* column, initially present only for non-hospitalised patients, was recalculated for all patients to ensure consistency across the dataset. This was done by dividing the values in the *KLC (mg/l)* column by those in the *LLC (mg/l)* column. Special attention was given to handling instances of division by zero, with infinite results being replaced by NaN.

5.2.4 Date Formats

A significant issue was encountered with the *DOB* column, which contained a variety of date formats such as "dd-mmm-yyyy", "dd/mm/yyyy", and "yyyy-mm-dd hh:mm", which led to misinterpretation and analysis difficulties. This variety led to misinterpretation and analysis difficulties. To address this, a systematic approach was taken to standardise all dates.

The first step was converting dates without specifying a format, falling back on several predefined formats if the initial attempt failed. After conversion, any time components were removed to ensure uniformity.

The standardised dates replaced the original entries, and the original column was dropped in favour of the new, consistently formatted column. This process enhanced the consistency and usability of the date information, facilitating more accurate and effective analysis.

Similarly, the *Date of Admission* column, crucial for calculating patients' ages, was also standardised. For patients who were not hospitalised (and thus had missing admission dates), a default date of 30th April 2024 was used. This date was chosen as it marked the dataset cut-off, and its impact on age calculations was minimal. This approach ensured continuity in age calculation, allowing the *Age* variable to be consistently derived from the birth and admission dates.

By standardising the *DOB* and *Date of Admission* columns, the dataset's consistency and accuracy were improved, facilitating more reliable analysis.

5.2.5 Comorbidities

The *Co-morbidities* column required significant preprocessing to ensure data were usable for analysis. This column contained various health conditions associated with each patient, often in inconsistent and mixed formats. For instance, the term "diabetes" appeared in multiple forms such as "diabetes type 2", "dm", or "t2dm".

First, missing values in the *Co-morbidities* column were replaced with empty strings to avoid issues during further processing. Each condition was then mapped to a broader disease category using a predefined dictionary that associated specific conditions with categories like "Cardiovascular Diseases", "Mental Health Disorders", and "Endocrine and Metabolic Disorders."

To create this mapping, a thorough review of each comorbidity was conducted by consulting online medical resources to understand each condition's nature and appropriate categorisation, ensuring that each condition was accurately mapped to its respective category, accounting for the diversity of used medical terminologies. The full mapping of each comorbidity to its corresponding category is detailed in Appendix A – *Comorbidities Categorisation*.

A function was implemented to map each condition to its respective category, handling multiple conditions within a single entry by splitting and cleaning the text. Conditions that could belong to multiple categories, such as "mixed hyperlipidaemia type 2 diabetes mellitus", were assigned to their respective categories. After categorisation, the column was *one-hot encoded*, creating binary columns indicating each category's presence or absence for every patient.

This structured approach ensured that the data in the *Co-morbidities* column was consistent and ready for analysis. This allowed for a comprehensive understanding of the various health conditions within the patient population and facilitated more precise insights during subsequent stages of analysis.

5.2.6 Postcodes

The *Postcode* column was first validated to ensure conformity to standard postcode formats using *regular expressions*. This format includes an outward code, identifying the broader delivery area, and an inward code, used for more precise sorting within that area. For example, in the postcode "PO1 3AX", "PO1" refers to the Portsmouth area and district, while "3AX" specifies the sector and individual delivery points (Education and Skills Funding Agency, 2017). Postcodes that did not match this format were identified and reviewed for inconsistencies.

To standardise the postcodes, a function was applied that corrected common formatting issues, such as unnecessary characters and spacing. Postcodes not recognised by the function were initially flagged for review and, after further investigation, identified as no longer in use. The obsolete postcodes were then researched online to determine their coordinates, ensuring accurate referencing. For instance, postcodes in the list starting with "CF4" were confirmed as obsolete and had their respective coordinates provided (Kane Data Limited, 2024; StreetCheck, 2024).

Geocoding was performed using the Nominatim geocoder to retrieve latitude and longitude for each postcode. Nominatim leverages OpenStreetMap data to locate places on Earth based on names and addresses (geocoding).[21] A cache was utilised to efficiently store and retrieve coordinates, streamlining the process. Distances from a reference postcode, "CF14 4XW" (University Hospital of Wales), were calculated for each entry.

To handle missing distance values, the median distance was used to fill in gaps, ensuring completeness and consistency in the dataset. The median, which is the middle value in a sorted list of numbers, was chosen over the mean because it is less affected by outliers (values significantly different from the rest) and skewed data (values are unevenly distributed). This approach helps maintain the representativeness of the imputed values.

[21] https://nominatim.org/

5.2.7 PP Type

The initial steps to clean and standardise the *PP Type* column involved replacing specific substrings and addressing formatting issues, following the guideline in Section 2.1.1 *Types of Multiple Myeloma*. This process aimed to generalise categories by combining similar ones into broader groups. This approach was adopted to reduce the total number of categories and increase the number of patients within each category. By doing so, the dataset benefits from improved statistical power, as larger sample sizes within each category enhance the reliability of statistical estimates. This results in more robust and generalisable insights from the analysis.

Substrings such as “band 1” and “band 2” were removed, and extra commas and spaces were addressed. Values were converted to lowercase for consistency. A dictionary of replacement mappings was used to handle various cases, such as converting “fllc only” to “fllc”, following the guideline described in Section 2.1.1. For multi-value cases, entries were split into lists using newline characters. To facilitate analysis, the cleaned column was one-hot encoded, which involved flattening lists into individual rows, applying one-hot encoding, and aggregating the results back into the original data frame.

5.2.8 Number of Cycles and Line of Treatment

Values in the *Number of cycles* column were first converted to strings and stripped of ordinal suffixes such as “th” (e.g., “5th” was simplified to “5”). These cleaned values were then converted to integers to ensure consistency. A similar process was applied to the *Line of treatment* column, where entries like “30th cycle Lenalidomide + Dexamethasone” were reduced to “30” to retain only the relevant treatment number, as the detailed treatment information is recorded separately in the Treatment column. The cleaned data were converted to integers for uniformity.

5.2.9 IgA and IgM Values

A function was defined to handle values in the *IgA (g/l)* and *IgM (g/l)* columns that started with “<”. For these values, the function calculated 75% of the given number (e.g., “<0.5” was adjusted to 0.375). This adjustment provided a more accurate representation of the data. Additionally, any “nan” as strings were converted to actual NaN values, and selected columns were converted to numeric types to ensure consistency across the dataset.

5.2.10 Removing Duplicate Records

Duplicates in the dataset were addressed by keeping only the first hospitalisation record for each patient. The DataFrame was sorted by *Patient Number* and *Date of Admission*, and duplicates were dropped, retaining the earliest admission record.

5.2.11 Gender

A function was applied to standardise the *Gender* column, consolidating “M” and “Male” values into “M”, and “F” and “Female” values into “F”.

5.2.12 Ethnicity

A function was implemented to standardise the *Ethnicity* column by consolidating similar categories into broader groups. This process aimed to simplify the dataset and address the issue of sparse representation in many categories, as most records were concentrated in the “Caucasian” category. The mapping was carried out as follows:

- Caucasian Europe: Entries such as “Caucasian (Italian)”, “Caucasian/Italian”, and “Caucasian/Polish” were grouped under “Caucasian Europe”.
- Caucasian UK: The generic term “Caucasian” was standardised to “Caucasian UK” to distinguish it from the European subcategories.
- South Asian: Ethnicities like “Indian”, “Bangladeshi”, and “Sinhalese” were combined into “South Asian”.
- African: Entries such as “Sudanese”, “African”, and “Afrocarribbean” (sic) were consolidated under “African”.

5.2.13 Handling Residual Missing Values

After all the procedures detailed in the previous sections, a small number of missing values remained in the columns *IgA (g/l)*, *IgM (g/l)*, *IgG (g/l)*, *KLC (mg/l)*, *LLC (mg/l)*, *PP*, *Neuts*, *Number of cycles*, *Ratio*, and *Age*. Given the minimal number of these residual missing values, the mean of each column was used to replace them, helping maintain the overall distribution and central tendency of the data. Since the missing values are few, this approach reduces potential biases and preserves the consistency of the dataset, ensuring that the analysis remains reliable and representative.

5.2.14 Final Data Cleaning and Column Removal

Columns not common to both original datasets or with excessive missing values were removed. This included columns with 95 or more NaNs or empty rows.

Overall, by meticulously identifying and correcting errors through automated scripts and standardising entries, the dataset was ensured to be robust and reliable. This foundational work enabled more accurate and insightful exploration and analysis, ultimately contributing to the reliability and validity of the study's findings.

6. Explore

This chapter presents a descriptive analysis of the hospitalisation data and associated medical and demographic factors, offering an overview of the dataset. The analysis began with an exploratory phase, where patterns and trends within the dataset were examined. This phase involved a detailed investigation of the hospitalisation records and associated variables, followed by a quantitative analysis where relationships and variations between key factors were contrasted. The characteristics of the data were carefully scrutinised, setting the stage for more in-depth analyses.[22]

6.1 Summary Statistics

The analysis began with a examination of the dataset's summary statistics, to get insights into the central tendencies, variability, and range of key variables. This quantitative overview forms the foundation for deeper analysis and interpretation. Here are the key findings:

- Hospitalisation: About 27% of the patients have been hospitalised; the rest didn't require hospital stays.
- Age: The patients have an average age of 69.8 years, with ages spanning from 23 to 91 years. This broad age range suggests that the dataset encompasses a wide spectrum of adults, from younger to elderly patients.
- Gender: The gender distribution is slightly male-biased, with 74 male patients compared to 57 female patients.
- Ethnicity: The dataset is predominantly composed of Caucasian UK patients (117), with much smaller representations from African (5), South Asian (5), and Caucasian European (4) backgrounds. The underrepresentation of minority groups may limit the generalisability of findings related to ethnicity and underscores the importance of considering these demographic factors in any subsequent analysis.
- Distance from UHW: The average distance of patients from the University Hospital of Wales (UHW) is 10.08 km, with some patients living as far as 87.38 km away. This geographic distribution may affect access to care and could be a factor in treatment decisions and outcomes.

[22] See Appendix F – *Descriptive Analysis*.

- Line of Treatment: The data show that most patients are undergoing their first line of treatment, with 61 individuals in this category. The average number of treatment lines across the dataset is 2.11, with the number of treatments ranging from 1 to 6. This indicates a broad spectrum of treatment histories, with fewer patients advancing to later lines of therapy.
- Number of Treatment Cycles: Patients undergo a wide range of treatment cycles, with an average of 14.6 cycles, a minimum of 1 and a maximum of 85.
- Health Conditions: The occurrence rate of specific health conditions, such as Skin Disorders, Endocrine and Metabolic Disorders, and Cardiovascular Diseases, is generally low. For example, Skin Disorders have a mean prevalence of 0.11, meaning they appear in 11% of the cases within the dataset, indicating that such conditions are relatively uncommon.
- IgG, IgA, IgM: The dataset captures significant variability in immunoglobulin levels, reflecting the diverse immune profiles of the patient population. The average IgG level is 7.74 g/l, with levels reaching as high as 53.97 g/l. Similarly, IgA and IgM levels show considerable spread, highlighting the heterogeneity in patients' immune responses.
- KLC, LLC: KLC and LLC present average values of 188.62 mg/l and 155.21 mg/l, and their ratio averages 0.23. These measurements provide insight into the relative abundance of these light chains, which can be relevant for understanding underlying pathological conditions in patients.
- Neutrophils (Neuts): The average neutrophil count is approximately 4.57, with a minimum value just above 0 and a maximum recorded value of 21.30. This variation underscores differences in immune function among patients, which may correlate with their response to treatment or disease progression.
- Specific Condition Indicators:
    - β2-microglobulin (b2m): The presence of β2-microglobulin is rare in this dataset, with an average value close to zero (mean of 0.0076).
    - Immunoglobulin Heavy Chains (IgA, IgD, IgG, IgM): These indicators reflect the presence of specific immunoglobulin heavy chains. Among them, IgA is the most frequently observed (mean of 0.1832), while IgD and b2m are the least common, each with a mean of approximately 0.0076.

- Light Chain: This condition, associated with light chain disease, appears in about 11.45% of the patients, indicating its relevance in this population.
- Non-Secretory Myeloma: Similar to b2m and IgD, this condition is quite rare, with a mean value of 0.0076, indicating that it is uncommon in the dataset.

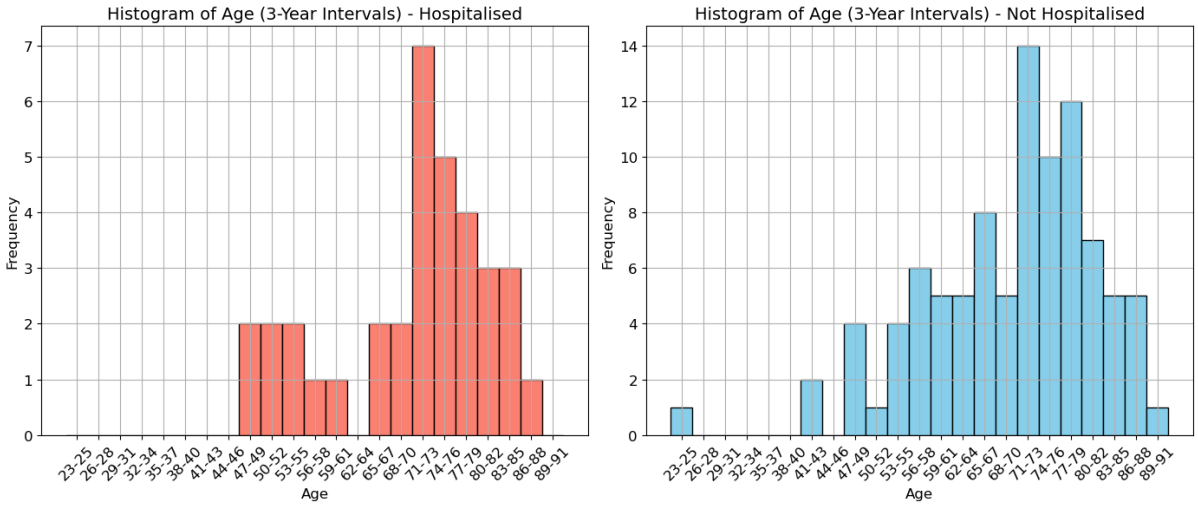
These summary statistics provide a comprehensive view of the dataset, highlighting key demographic and health-related characteristics. This overview serves as a crucial baseline for further analysis, enabling the identification of trends, correlations, and patterns related to patient health and treatment outcomes. The variability observed in numerical and categorical variables underscores the need for tailored approaches in the analysis, considering the diverse nature of the patient population.

6.2 Age Distribution by Hospitalisation Status

To explore the age distribution among hospitalised and non-hospitalised patients, histograms were generated using 3-year intervals (see Figure 3). This exploration was conducted to gain a deeper understanding of the dataset and the characteristics of the patients, providing context for subsequent analyses.

In both plots, the x-axis represents age intervals, and the y-axis shows the number of patients within each interval.

Figure 3 *Histograms of Age*

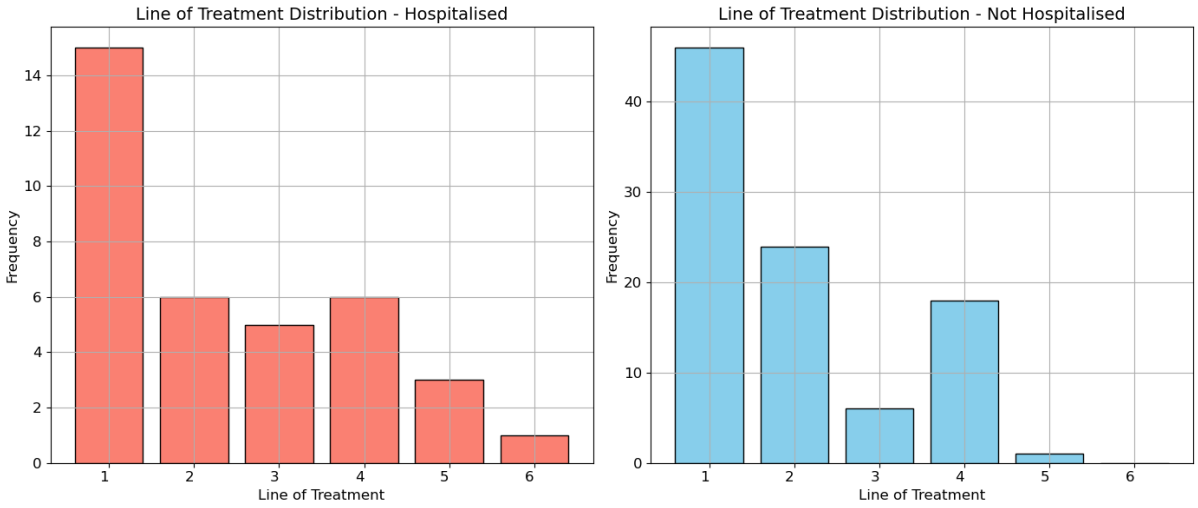


These visualisations reveal patterns in age distribution relative to hospitalisation status, potentially identifying age groups with higher hospitalisation rates.

6.3 Line of Treatment Distribution by Hospitalisation Status

Figure 4 illustrates the distribution of treatment lines between hospitalised and non-hospitalised patients. All possible treatment lines are included to ensure consistency across both plots, with missing categories in each group filled in as zero.

Figure 4 *Line of Treatment Distribution*

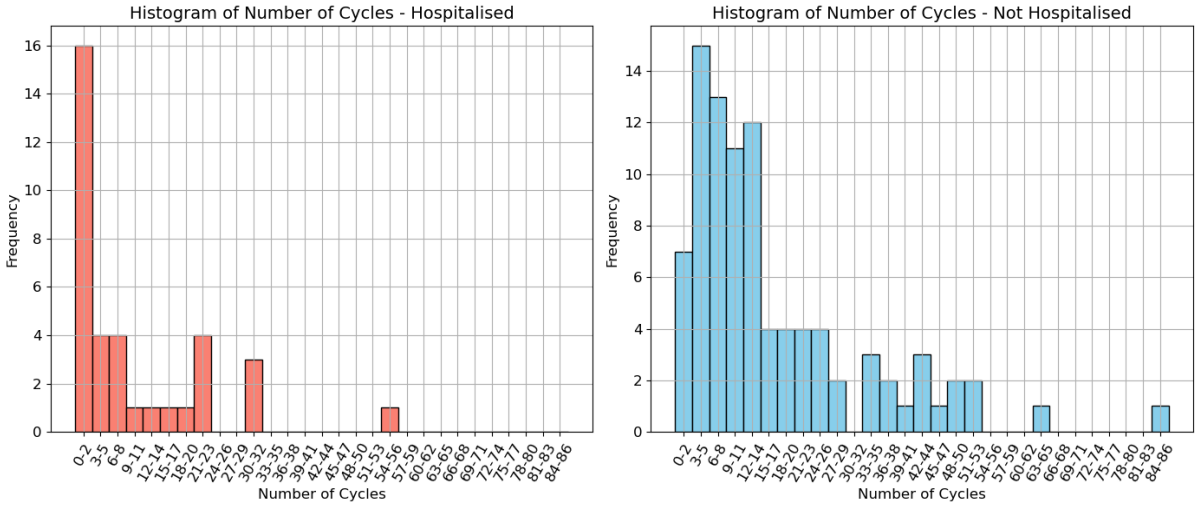


Although the overall distribution of treatment lines is similar between the two groups, a notable observation is that a higher proportion of patients in the later lines of treatment are hospitalised. This is evident from the comparable numbers in later treatment lines, despite the total number of hospitalised patients being significantly smaller than that of non-hospitalised patients.

6.4 Number of Cycles Distribution by Hospitalisation Status

Figure 5 illustrates the distribution of the number of treatment cycles for hospitalised and non-hospitalised patients. The bins used in the histograms span the range of treatment cycles, with intervals of 3 cycles.

Figure 5 *Number of Cycles Distribution*

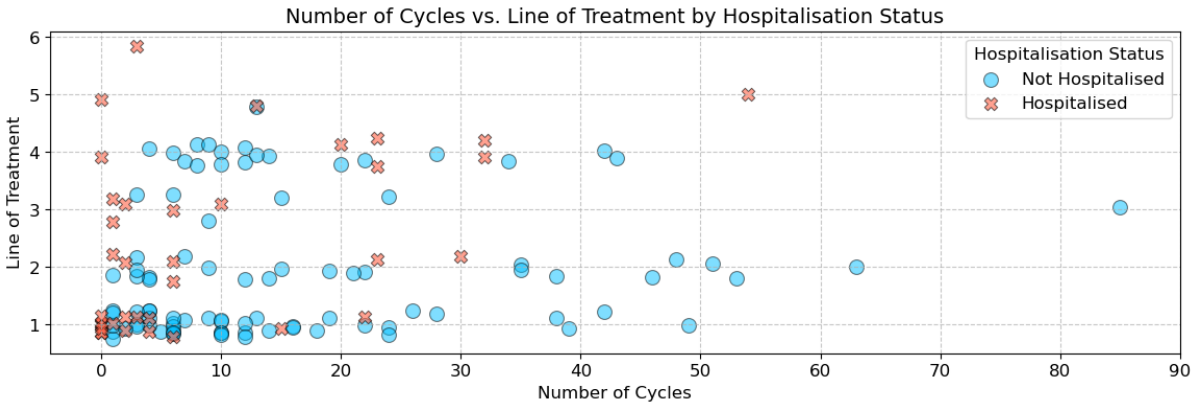


The data reveal that hospitalised patients tend to have fewer treatment cycles. In contrast, the histogram for non-hospitalised patients shows a higher frequency of patients with a greater number of cycles. This suggests that patients with more treatment cycles are less likely to be hospitalised, indicating a potential trend in the treatment duration and hospitalisation status.

6.5 Number of Cycles vs. Line of Treatment by Hospitalisation Status

Figure 6 presents a scatter plot that illustrates the relationship between hospitalisation status, the number of treatment cycles, and the lines of treatment. A *vertical jitter* has been applied in the graph to enhance the visibility of overlapping data points.

Figure 6 *Number of Cycles vs. Line of Treatment by Hospitalisation Status*



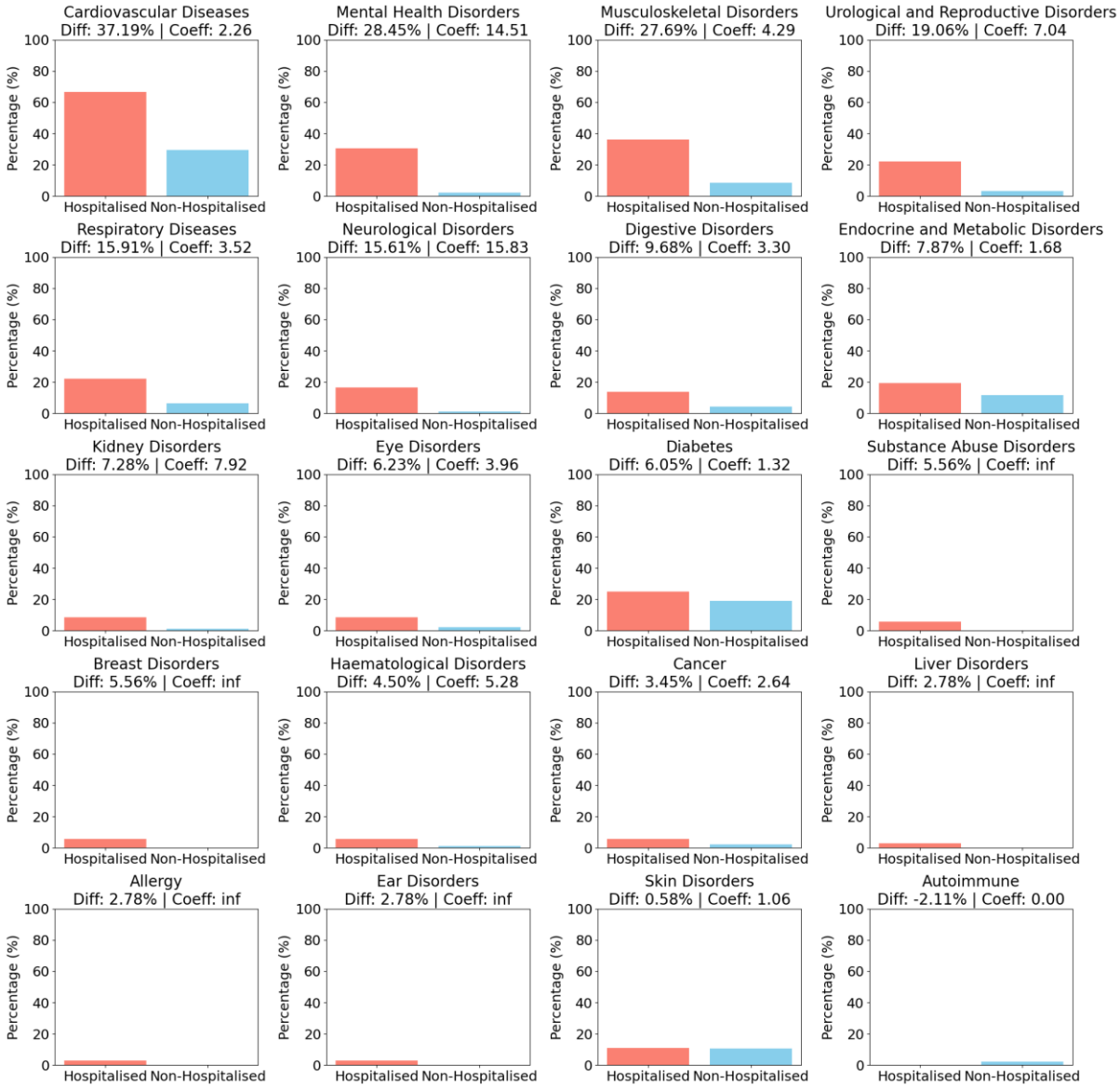
As observed in the previous sections, Figure 6 shows a discernible pattern regarding hospitalisation status in relation to the number of treatment cycles and lines of treatment. Patients who have undergone a greater number of cycles – indicating a longer duration of

treatment – tend to be less frequently hospitalised. This suggests that extended treatment durations do not necessarily correlate with increased hospitalisation. Conversely, patients in higher lines of treatment are more likely to be hospitalised. This pattern may reflect the progression of the disease; those in later treatment lines might have been managing MM for a longer period, potentially leading to more severe disease manifestations and, consequently, higher hospitalisation rates.

6.6 Health Disorders Distribution by Hospitalisation Status

Figure 7 illustrates the prevalence of various health disorders among hospitalised and non-hospitalised patients.

Figure 7 *Health Disorders Distribution*



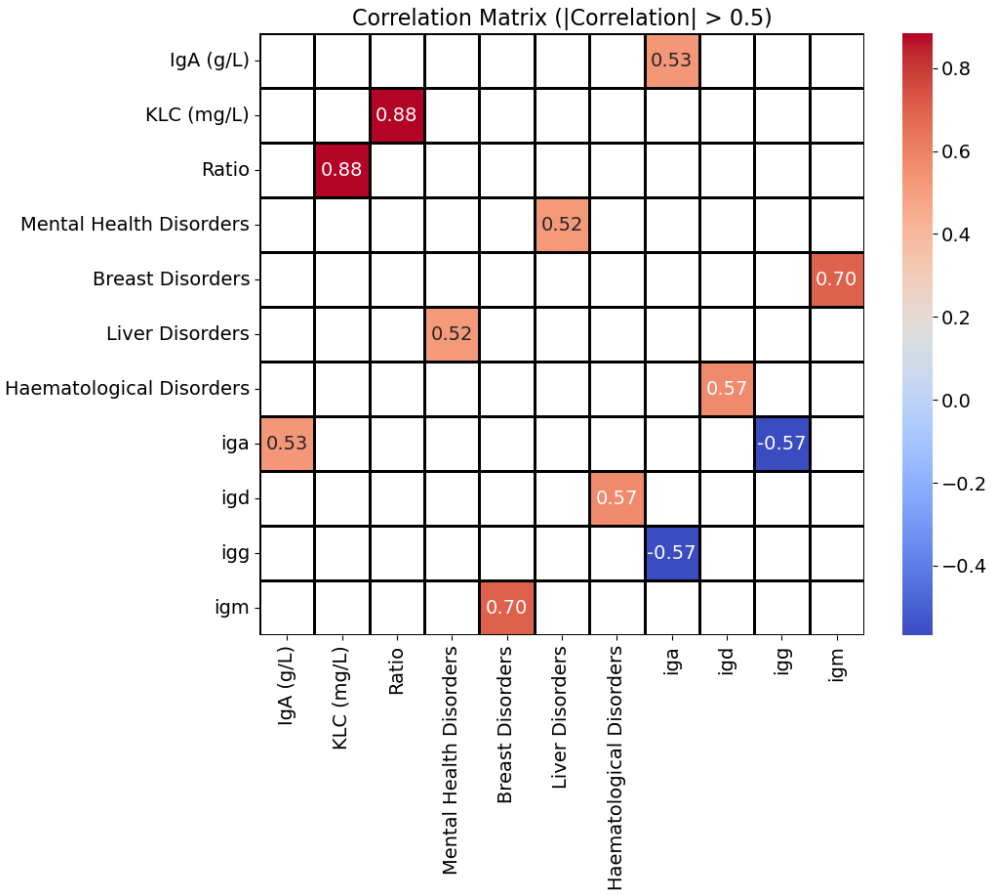
Each subplot shows the percentage of patients with a particular disorder in both groups, with differences and coefficients displayed in the titles.

Notably, the first row of plots, which includes Cardiovascular Diseases, Mental Health Disorders, Musculoskeletal Disorders, and Urological and Reproductive Disorders, appears to highlight the most significant differences between the hospitalised and non-hospitalised groups. These disorders stand out as the most relevant in terms of their association with hospitalisation status, suggesting they may have a stronger impact on the need for hospital care.

6.7 Correlation Analysis

The filtered correlation matrix presented in Figure 8 highlights a few moderate relationships among variables.

Figure 8 *Correlation Matrix of Strongest Correlations*



Notably, Breast Disorders and IgM (g/l) have a correlation of 0.70, suggesting a potential link. Haematological Disorders and IgD (g/l) also show a positive correlation of 0.57.

Additionally, IgA (g/l) correlates with both IgA (0.56) and IgG (-0.53). Liver Disorders and Mental Health Disorders have a moderate correlation of 0.52. These findings suggest some interactions between these variables, though most relationships remain relatively weak. As expected, due to the Ratio being a calculation based on KLC, these two variables are highly correlated, reflecting their inherent mathematical relationship.

7. Model

This chapter analyses hospitalisation risk among MM patients through statistical and predictive modelling techniques. The first part focuses on the use of the two-sample t-test to examine differences in specific clinical characteristics (number of treatment cycles and treatment lines) between hospitalised and non-hospitalised patients. These analyses offer a detailed comparison of these groups, shedding light on the potential impact of hospitalisation on treatment intensity and progression.

The second part shifts towards predictive modelling, employing logistic regression to identify and predict key risk factors for hospital admission. The model is refined and evaluated to enhance its precision and recall in predicting hospitalisation risk. By integrating these methodologies, the chapter not only highlights significant hospitalisation determinants but also develops a predictive tool to assist in clinical decision-making. This approach provides valuable insights into understanding and mitigating hospitalisation risks in MM patients.

7.1 Two-sample T-Test

This chapter explores the use of the two-sample t-test in the project. As highlighted in Section 2.2.1 Key Statistical Techniques, it is a robust method for comparing the means of two independent groups. It is particularly effective for situations where data points are not paired or matched, such as comparing measurements between hospitalised and non-hospitalised MM patients (Ross & Willson, 2017).

In this project, it is used to evaluate whether there are significant differences in variables between these two groups of MM patients,[23] facilitating the assessment of mean differences in selected variables between the observations of hospitalised and non-hospitalised patients.[24]

7.1.1 Comparison of Treatment Cycles by Hospitalisation Status

This section presents the results of a one-tailed independent samples t-test conducted to assess whether the mean number of treatment cycles differs significantly between hospitalised and non-hospitalised MM patients. Specifically, the test aimed to determine if the

[23] This section answers the first question presented in Section 1.4 *Research Questions*: *Are patients more or less likely to be hospitalised as they advance in their treatments, considering factors such as treatment cycles and lines of therapy?*
[24] See Appendix G – *Paired T-Tests*.

mean number of cycles for the hospitalised group was less than that for the non-hospitalised group.

The analysis was performed by comparing the hospitalised group (MM patients who have been hospitalised) with the non-hospitalised group (MM patients who have not been hospitalised). The one-tailed independent samples t-test, assuming unequal variances between the two groups (Welch's t-test), was used with the following hypothesis:

- Null Hypothesis (H0): The mean number of treatment cycles for the hospitalised group is equal to or greater than that of the non-hospitalised group.
- Alternative Hypothesis (H1): The mean number of treatment cycles for the hospitalised group is less than that of the non-hospitalised group.

The results of the t-test are as follows:

- T-statistic: -2.5496
- P-value: 0.0064

At the 99% confidence level, with a significance threshold (alpha) of 0.01, the p-value of 0.0061 is below this threshold. Consequently, the null hypothesis is rejected with 99% confidence, leading to the conclusion that the mean number of treatment cycles for the hospitalised group is significantly smaller than that of the non-hospitalised group. This result suggests that, on average, hospitalised patients have undergone fewer treatment cycles than those who have not been hospitalised. Such a finding may reflect underlying differences in treatment management or disease progression between the two groups.

7.1.2 Comparison of Treatment Lines by Hospitalisation Status

This section presents the results of a two-sided independent samples t-test conducted to assess whether there is a significant difference in the mean number of treatment lines between hospitalised and non-hospitalised MM patients. The test aimed to determine if there is a difference, in either direction, between the mean number of treatment lines for the hospitalised group and the non-hospitalised group.

The analysis was performed by comparing the hospitalised group with the non-hospitalised group. The two-sided independent samples t-test, assuming unequal variances between the two groups (Welch's t-test), was used with the following hypotheses:

- Null Hypothesis (H0): There is no difference in the mean number of treatment lines between the groups.
- Alternative Hypothesis (H1): There is a difference in the mean number of treatment lines between the groups.

The results of the t-test are as follows:

- T-statistic: 1.5201
- P-value: 0.1345

At the 95% confidence level, with a significance threshold (alpha) of 0.05, the p-value of 0.135 is above this threshold. Consequently, the null hypothesis is not rejected. This indicates there is no significant evidence to suggest a difference in the mean number of treatment lines between the hospitalised and non-hospitalised groups. This result suggests that, on average, there is no significant variation in the number of treatment lines experienced by patients, regardless of their hospitalisation status.

7.2 Logistic Regression

This section focuses on the building of the logistic regression model, chosen for its interpretability and ability to identify relationships between predictors and outcomes. A random *train/test split* was applied, with 70% of the data used for training and 30% for testing, for an unbiased evaluation. Key performance metrics, including accuracy, F1 score, precision, and recall, were calculated, and a confusion matrix was used to summarise the results visually. All evaluations were based on the test set to assess how well the model generalises to unseen data.

The analysis began by preparing the data, focusing on the target variable, hospitalisation status, while excluding non-relevant columns such as *Patient Number* (unique identifier so not relevant for analysis) and *Line of Treatment Jittered* (jittered version of *Line of treatment* column). Notably, *Ethnicity* was excluded due to its dominance of one type and minimal representation of others, as observed in Section 6.1.

Categorical variables, such as *Gender*, were one-hot encoded to transform them into a format suitable for the logistic regression model.

7.2.1 Hospital Admission Risk Factors Identification

This section outlines the logistic regression model developed to identify risk factors for hospital admission in MM patients.[25] [26]

An initial logistic regression model was trained using all available features. Then, features appearing fewer than five times were excluded to improve reliability. Highly correlated variables related to IgG, IgA, and IgM were removed due to their limited predictive value, as noted in Section 6.7. Features with absolute coefficients below 0.3 were also excluded, as this threshold, representing an odds ratio of just over 1.3 (or below 0.8 if the coefficient were negative), ensured only meaningful predictors were retained.

Using the top 10 most impactful features, the refined model achieved 85% accuracy on the test set. The classification report and confusion matrix are detailed below.

Table 1 *Classification Report*

Class	Precision	Recall	F1-Score	Support
0 (Non-Hospitalised)	0.83	1.00	0.91	29
1 (Hospitalised)	1.00	0.45	0.62	11
Accuracy Macro Average	0.69	0.6	0.77	40
Accuracy Weighted Average	0.72	0.75	0.83	40

Table 2 *Confusion Matrix*

	Predicted: Non-Hospitalised (0)	Predicted: Hospitalised (1)
Actual: Non-Hospitalised (0)	29	0
Actual: Hospitalised (1)	6	5

Moreover, the logistic regression model identifies several key features associated with hospital admission. The following table summarises these features along with their coefficients, absolute coefficients, odds ratios, and non-zero counts:

[25] See Appendix H – *Logistic Regression for Hospital Admission Risk Factors*.
[26] This section answers the second question presented in Section 1.4 *Research Questions*: *What demographic, clinical, and treatment factors contribute to the risk of hospital admission during MM treatment?*

Table 3 *Predominant Risk Factors*

Feature	Coefficient	Absolute Coefficient	Odds Ratio	Occurrences
Mental Health Disorders	2.04	2.04	7.73	12
Musculoskeletal Disorders	1.49	1.49	4.44	19
Cardiovascular Diseases	1.37	1.37	3.92	47
Respiratory Diseases	1.08	1.08	2.94	13
Neurological Disorders	0.97	0.97	2.64	6
Diabetes	0.92	0.92	2.51	27
Digestive Disorders	0.89	0.89	2.43	9
Urological and Reproductive Disorders	0.71	0.71	2.04	9
Endocrine and Metabolic Disorders	0.50	0.50	1.65	17
Line of Treatment	0.31	0.31	1.36	131

The model highlights the following key features:

- Mental Health Disorders: This feature has the highest coefficient of 2.04 and an odds ratio of 7.73. It indicates that patients with mental health disorders are almost 8 times more likely to be hospitalised. This substantial coefficient underscores the significant impact of mental health on hospitalisation risk.
- Musculoskeletal Disorders: With a coefficient of 1.49 and an odds ratio of 4.44, this feature suggests that patients with musculoskeletal disorders are over four times more likely to be hospitalised, highlighting its considerable role in predicting hospitalisation.
- Cardiovascular Diseases: This feature has a coefficient of 1.37 and an odds ratio of 3.92, indicating that patients with cardiovascular diseases are nearly four times more likely to be hospitalised, underscoring its importance as a predictor of hospital admission.
- Respiratory Diseases: With a coefficient of 1.08 and an odds ratio of 2.94, respiratory diseases are also a significant risk factor, although with a somewhat lower impact than the top three features.
- Neurological Disorders: This feature has a coefficient of 0.97 and an odds ratio of 2.64, demonstrating a notable association with hospitalisation risk.

These findings highlight the most influential factors associated with an increased likelihood of hospitalisation. Given this, addressing these conditions could be critical for reducing hospital admissions.

7.2.2 Hospitalisation Prediction

Building on the previous model, this section focuses on refining the logistic regression model's predictive capability for hospitalisation risk among MM patients.[27] [28] The goal was to enhance the model's ability to identify patients at high risk of hospitalisation, with a particular emphasis on balancing precision, recall, and F1 score. Among these metrics, the F1 score was prioritised as the primary criterion for model optimisation due to its ability to provide a balanced assessment of the model's performance across both precision and recall. This balance is crucial in scenarios where both false positives and false negatives carry significant consequences.

The refinement process emphasised three key performance metrics:

- Precision for the Positive Class: Precision measures the proportion of true positive predictions among all positive predictions made. Maintaining a high precision to minimise false positives was important, thereby avoiding unnecessary treatments and interventions for patients incorrectly predicted to be at risk of hospitalisation.
- Recall for the Positive Class: Recall (or sensitivity) measures the proportion of actual positives (hospitalised patients) correctly identified by the model. While precision was important, ensuring that the model correctly identified a significant proportion of hospitalised cases was also crucial.
- F1 Score: The F1 score was given the most weight in model selection because it balances precision and recall, providing a single metric that reflects the model's overall effectiveness. This was especially useful in a medical context, where both types of errors—missing a patient who should be hospitalised and falsely identifying a patient as at risk—can have serious consequences.

To optimise the model, *hyperparameter tuning* was conducted, resulting in the following optimal parameters:

[27] See Appendix I – *Logistic Regression for Enhanced Prediction*.
[28] This section answers the third question presented in Section 1.4 *Research Questions*: *How effective is a predictive risk model in identifying MM patients at higher risk of hospitalisation?*

- Best Parameters: C = 10,[29] penalty = l1,[30] solver = liblinear[31]
- Best F1 Score: 0.8222

The model was then evaluated on the test set, producing the results below:

Table 4 *Classification Report – Optimised Model*

Class	Precision	Recall	F1-Score	Support
0 (Non-Hospitalised)	0.88	0.97	0.92	29
1 (Hospitalised)	0.88	0.64	0.74	11
Accuracy Macro Average	0.88	0.80	0.83	40
Accuracy Weighted Average	0.88	0.88	0.87	40

Table 5 *Confusion Matrix – Optimised Model*

	Predicted: Non-Hospitalised (0)	Predicted: Hospitalised (1)
Actual: Non-Hospitalised (0)	28	1
Actual: Hospitalised (1)	4	7

Table 6 *Model Evaluation Metrics – Optimised Model*

Metric	Value
Accuracy Score	0.8750
F1 Score	0.7368
Precision	0.8750
Recall	0.6364

These results show that the refined model effectively balances precision and recall, with the F1 score providing a robust measure of its overall performance. Prioritising the F1 score ensured reliable identification of hospitalisation risk, making the model a useful tool for clinical decisions. High precision helps avoid unnecessary treatments, while the model's

[29] C = 10: This parameter, cost function, controls the inverse of regularisation strength. A larger C value, such as 10, reduces regularisation, allowing the model to fit the data more closely. While this can improve accuracy, it also raises the risk of overfitting by making the model more flexible. Therefore, fine-tuning C is crucial for balancing adaptability and generalisation, ensuring maximum performance of the model (Arafa, Radad, Badawy, & El-Fishawy, 2022).
[30] Penalty = L1: This indicates the use of L1 regularisation (lasso), which promotes sparsity by penalising the absolute values of the coefficients. This approach causes the model to shrink some coefficients to zero, effectively selecting only the most important features and excluding irrelevant ones. To prevent overfitting, L1 regularisation adds a penalty term to the cost function, enhancing the model's ability to generalise by discouraging overly complex solutions (Arafa, Radad, Badawy, & El-Fishawy, 2022).
[31] Solver = liblinear: The liblinear solver is well-suited for smaller datasets and L1 regularisation. It uses a coordinate descent algorithm to iteratively adjust parameters and minimise the cost function, ensuring accurate model fitting (scikit-learn developers, 2024).

recall shows it correctly identifies 63.6% of actual hospitalised cases. However, there is still room to improve sensitivity. Overall, the model represents a step forward in predicting hospitalisation risk for MM patients, though the lower recall suggests further refinement is needed to enhance sensitivity and predictive power.

8. Interpret

The analysis conducted in the preceding chapters, particularly the models developed and refined in Section 7.2, provides insights into the factors influencing hospitalisation risk among MM patients.

The model outlined in Section 7.2.1, identified several key predictors of hospitalisation, including Mental Health Disorders, Musculoskeletal Disorders, and Cardiovascular Diseases. These features demonstrated the strongest association with the likelihood of hospital admission. This suggests that patients with these conditions are more likely to require hospitalisation, highlighting the importance of these health issues in the management of MM patients.

The refined (final) model in Section 7.2.2 focused on optimising predictive performance, particularly by tuning the model's hyperparameters to enhance its ability to correctly identify hospitalised patients. This process included a rigorous evaluation of the model's precision, recall, and particular emphasis on the F1 score, which offers a comprehensive measure of the model's effectiveness in predicting hospitalisation, accounting for both false positives and false negatives.

The final model achieved an F1 score of 0.7368 on the test set, reflecting its improved capability to predict hospitalisation accurately. The associated precision and recall metrics also indicated that the model struck a reasonable balance between avoiding false alarms and correctly identifying patients at risk of hospitalisation.

These results underscore the significance of considering multiple aspects of a patient's health status when predicting hospitalisation risk. While the initial model highlighted the importance of specific health conditions, the refined model's performance demonstrated that a more nuanced approach, including careful model tuning and feature selection, can lead to better predictive accuracy.

Overall, the investigation into hospitalisation risk highlights the complexity of predicting such outcomes and the need for models that not only identify key risk factors but also accurately balance the trade-offs between different metrics. This analysis provides a foundation for further research and potential clinical application, offering insights that could contribute to improved patient care and resource allocation in healthcare settings.

9. Conclusions and Future Work

The insights gained from this study have provided a strong foundation for understanding the factors influencing the hospitalisation of MM patients. By developing a logistic regression model, key predictors of hospital admissions were identified, and the model's performance was fine-tuned to deliver balanced and accurate predictions. However, several promising opportunities for future research have emerged, offering potential pathways to enhance and expand upon these findings.

One of the most significant future opportunities lies in expanding the dataset. The current study utilised data from a single hospital, which, while valuable, may not fully capture the diversity of patient experiences and outcomes seen across different hospitals or regions. Research could benefit greatly from incorporating larger and more varied datasets. Such an approach would allow validating these findings on a broader scale, improving the model's generalisability and applicability.

Another important area for continued research is the deeper examination and mitigation of potential biases. While efforts were made to address data limitations, the dataset's origin from a single source could mean it might not fully capture the diversity of the broader patient population. The decision to exclude variables such as ethnicity, due to data imbalance, underscores the challenges of working with incomplete datasets. Also, hospitalisation decisions were not always consistent and could vary depending on the individual making the decision, introducing another layer of potential bias. Future work should build on this by enhancing data collection efforts to include a more diverse range of patient characteristics and by conducting comprehensive bias analyses. This will help refine the predictive models further and ensure their fairness and accuracy across all patient groups.

The data collection process itself presents another opportunity for enhancement. In this study, the data were manually entered, a method prone to inconsistencies and errors. A more streamlined and automated approach to data collection would be beneficial.

Exploring advanced modelling techniques and incorporating additional variables into the predictive model could also be worthwhile. As healthcare data continues to grow in volume and complexity, there are opportunities to explore machine learning methods that can handle large, multidimensional datasets and uncover more complex patterns.

In conclusion, while this study has laid important groundwork in identifying hospitalisation risk factors for MM patients and developing a predictive model, there are several avenues for future research that could further refine and expand these findings. By incorporating larger and more diverse datasets, addressing potential biases, improving data collection methods, and exploring advanced modelling techniques, future work can build on this study to ultimately improve patient care and more effectively manage hospitalisation risks among MM patients.

BIBLIOGRAPHY

Alexander, D. D., Mink, P. J., Adami, H.-O., Cole, P., Mandel, J. S., Oken, M. M., & Trichopoulos, D. (2007). Multiple myeloma: a review of the epidemiologic literature. *Int J Cancer, 120 Suppl 12*, 40-61. doi:10.1002/ijc.22718

Allegra, A., Tonacci, A., Sciaccotta, R., Genovese, S., Musolino, C., Pioggia, G., & Gangemi, S. (2022). Machine Learning and Deep Learning Applications in Multiple Myeloma Diagnosis, Prognosis, and Treatment Selection. *Cancers, 14*(3), 606. doi:10.3390/cancers14030606

Arafa, A. H., Radad, M., Badawy, M. M., & El-Fishawy, N. (2022). Logistic Regression Hyperparameter Optimization for Cancer Classification. *Menoufia Journal of Electronic Engineering Research, 31*(1), 1-8. doi:10.21608/mjeer.2021.70512.1034

Armitage, P., Berry, G., & Matthews, J. N. (2001). Clinical trials. In P. Armitage, G. Berry, & J. N. Matthews, *Statistical Methods in Medical Research* (4 ed., Vol. 25, p. 832). Wiley, 2001.

Bakken, S. (2019, March). The journey to transparency, reproducibility, and replicability. *Journal of the American Medical Informatics Association, 26*(3), 185-187. doi:10.1093/jamia/ocz007

Becker, N. (2011). Epidemiology of Multiple Myeloma. In T. Moehler, & H. Goldschmidt (Eds.), *Multiple Myeloma. Recent Results in Cancer Research, vol 183.* Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-85772-3_2

Benjamin, M., Reddy, S., & Brawley, O. (2003). Myeloma and race: A review of the literature. *Cancer Metastasis Rev, 22*, 87-93. doi:10.1023/A:1022268103136

Bhutani, M., Blue, B. J., Cole, C., Badros, A. Z., Usmani, S. Z., Nooka, A. K., . . . Mikhael, J. (2023). Addressing the disparities: the approach to the African American patient with multiple myeloma. *Blood Cancer J. 13*, 189. doi:10.1038/s41408-023-00961-0

Blood cancer UK. (n.d.). Retrieved July 2, 2024, from Myeloma: https://bloodcancer.org.uk/understanding-blood-cancer/myeloma/

Boateng, E. Y., & Abaye, D. A. (2019). A Review of the Logistic Regression Model with Emphasis on Medical Research. *Journal of Data Analysis and Information Processing, 7*(4), Paper ID 95655, 18 pages. doi:10.4236/jdaip.2019.74012

Brandt, P. S. (2016). *The emergence of the data science profession.* Columbia University. doi:10.7916/D8BK1CKJ

Brown, L. M., Gridley, G., Pottern, L. M., Baris, D., Swanson, C. A., Silverman, D. T., . . . Fraumeni, J. F. (2001). Diet and nutrition as risk factors for multiple myeloma among blacks and whites in the United States. *Cancer Causes Control, 12*, pp. 117-125. doi:10.1023/A:1008937901586

Cancer Research UK. (2023, October 13). *About Cancer*. Retrieved August 16, 2024, from Types of myeloma: https://www.cancerresearchuk.org/about-cancer/myeloma/types

Cancer Research UK. (2023, September 29). *About Cancer*. Retrieved from Myeloma: https://www.cancerresearchuk.org/about-cancer/myeloma

Chevening. (n.d.). *About Chevening*. Retrieved from Chevening: https://www.chevening.org/about/

Education and Skills Funding Agency. (2017). *Individualised Learner Record (ILR) Specification 2017 to 2018 – Appendix C – Valid postcode format – Version 1.* Education and Skills Funding Agency. Retrieved from https://assets.publishing.service.gov.uk/media/5a81ebbded915d74e6234d42/Appendix_C_ILR_2017_to_2018_v1_Published_28April17.pdf

GOV.UK. (n.d.). *National Institute for Health and Care Excellence*. Retrieved July 12, 2024, from https://www.gov.uk/government/organisations/national-institute-for-clinical-excellence

Guetterman, T. C. (2019). Basics of statistics for primary care research. *Fam Med Community Health, 7*(2), e000067. doi:10.1136/fmch-2018-000067

Hill, T. (2023, October 02). *Bias in AI for precision medicine.* Retrieved from The REPROCELL Blog: https://www.reprocell.com/blog/bias-in-ai-for-precision-medicine

Hill, T. (2023, October 31). *Data Privacy and Protection in AI for Precision Medicine.* Retrieved from The REPROCELL Blog: https://www.reprocell.com/blog/data-privacy-and-protection-in-ai-for-precision-medicine

Kane Data Limited. (2024). *Find that Postcode*. Retrieved from https://findthatpostcode.uk/

Kim, T. K. (2015, November 25). T test as a parametric statistic. *Korean Journal of Anesthesiology, 68*(6), 540-546. doi:10.4097/kjae.2015.68.6.540

Lao, R. (2017, November 6). *Life of Data | Data Science is OSEMN.* Retrieved from Medium: https://medium.com/@randylaosat/life-of-data-data-science-is-osemn-f453e1febc10

Lau, C. (2019, January 3). 5 Steps of a Data Science Project Lifecycle. *Towards Data Science*. Retrieved from https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Mackenzie, J., Sheldon, J., Morgan, G., Cook, G., Schulz, T., & Jarrett, R. (1997). HHV-8 and multiple myeloma in the UK. *RESEARCH LETTERS, 350*(9085), pp. 1144-1145. doi:10.1016/S0140-6736(05)63792-0

Mason, H. (2022, July 5). The OSEMN ("Awesome") Data Science Process. *Super Data Science: ML & AI Podcast with Jon Krohn*. (J. Krohn, Interviewer) Retrieved from https://www.superdatascience.com/podcast/narrative-ai-with-hilary-mason

Mason, H., & Wiggins, C. (2010). *A Taxonomy of Data Science.* Retrieved from https://introdatasci.dlilab.com/pdf/A_Taxonomy_of_Data_Science.pdf

Monteith, B. E., Sandhu, I., & Lee, A. S. (2023, April 22). Management of Multiple Myeloma: A Review for General Practitioners in Oncology. *Curr Oncol., 30*(5), 4382-4401. doi:10.3390/curroncol30050334

Moss, S. E., Klein, R., & Klein, B. E. (1999, September 27). Risk Factors for Hospitalization in People With Diabetes. *Arch Intern Med., 159*(17), 2053-2057. doi:10.1001/archinte.159.17.2053

MyelomaUK. (n.d.). *MyelomaUK*. Retrieved June 27, 2024, from UNDERSTANDING MYELOMA: https://www.myeloma.org.uk/

MyelomaUK. (n.d.). *UNDERSTANDING MYELOMA*. Retrieved from What is myeloma?: https://www.myeloma.org.uk/understanding-myeloma/what-is-myeloma/

NHS. (2021, June 2). *NHS*. Retrieved June 28, 2024, from https://www.nhs.uk/conditions/multiple-myeloma/

NICE. (2018, October 25). *Myeloma: diagnosis and management*. Retrieved from NICE Guidance: https://www.nice.org.uk/guidance/ng35

Palumbo, A., Avet-Loiseau, H., oliva, S., Oliva, S., Lokhorst, H. M., Goldschmidt, H., . . . Moreau, P. (2015, August 3). Revised International Staging System for Multiple Myeloma: A Report From International Myeloma Working Group. *Journal of Clinical Oncology, 33*(26), 2863-2869. doi:10.1200/JCO.2015.61.2267

Panos, G. D., & Boeckler, F. M. (2023). Statistical Analysis in Clinical and Experimental Medical Research: Simplified Guidance for Authors and Reviewers. *Drug Des Devel Ther., 17*, 1959-1961. doi:10.2147/DDDT.S427470

Patten, M. L. (2017). *Proposing Empirical Research* (5th Edition ed.). Routledge.

python.org. (2024, August 16). *python*. Retrieved from What is Python? Executive Summary: https://www.python.org/doc/essays/blurb/?external_link=true

Rajkumar, S., & Kumar, S. (2020). Multiple myeloma current treatment algorithms. *Blood Cancer J., 10*, 94. doi:10.1038/s41408-020-00359-2

Ries, L. A., Harkins, D., Krapcho, M., Mariotto, A., Miller, B. A., Feuer, E. J., . . . Edwards, B. (Eds.). (2006). SEER Cancer Statistics Review, 1975-2003. *National Cancer Institute*. Retrieved from https://seer.cancer.gov/csr/1975_2003/

Ross, A., & Willson, V. L. (2017). Independent Samples T-Test. In *Basic and Advanced Statistical Tests* (pp. 13–16). Rotterdam: SensePublishers. doi:10.1007/978-94-6351-086-8_3

scikit-learn developers. (2024). *scikit learn.* Retrieved September 5, 2024, from Linear Models: https://scikit-learn.org/stable/modules/linear_model.html

Sergentanis, T. N., Zagouri, F., Tsilimidos, G., Tsilimidos, A., Tsagianni, M., Dimopoulos, M. A., & Psaltopoulou, T. (2015). Risk Factors for Multiple Myeloma: A Systematic

Review of Meta-Analyses. *Clinical Lymphoma Myeloma and Leukemia, Volume 15, Issue 10*, Pages 563-577.e3. doi:10.1016/j.clml.2015.06.003

Singh, P., Singh, N., Singh, K. K., & Singh, A. (2021). Chapter 5 - Diagnosing of disease using machine learning. In K. K. Singh, M. Elhoseny, A. Singh, & A. A. Elngar (Eds.), *Machine Learning and the Internet of Medical Things in Healthcare* (pp. 89-111). Academic Press. doi:10.1016/B978-0-12-821229-5.00003-3

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427-437. doi:10.1016/j.ipm.2009.03.002

StreetCheck. (2024). *Postcode Information*. Retrieved from All Postcode Districts Covered by StreetCheck: https://www.streetcheck.co.uk/postcode/alldistricts

Su, T.-L., Jaki, T., Hickey, G. L., Buchan, I., & Sperrin, M. (2016, July 26). A review of statistical updating methods for clinical prediction models. *Statistical Methods in Medical Research, 27*(1), 185-197. doi:10.1177/0962280215626466

The American Cancer Society medical and editorial content team. (2024, January). *Multiple Myeloma*. Retrieved from Risk Factors for Multiple Myeloma: https://www.cancer.org/cancer/types/multiple-myeloma/causes-risks-prevention/risk-factors.html

The International Myeloma Foundation. (2021, June 6). *International Myeloma Foundation*. Retrieved August 16, 2024, from https://www.myeloma.org/types-of-myeloma

UCLA: Statistical Consulting Group. (2024). *UCLA Advanced Research Computing: Statistical Methods and Data Analytics*. Retrieved September 5, 2024, from FAQ: What are the differences between one-tailed and two-tailed tests?: https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-what-are-the-differences-between-one-tailed-and-two-tailed-tests/

Watson, R. (2015). Quantitative research. *Nursing standard : official newspaper of the Royal College of Nursing, 29*(31), 44-48. doi:10.7748/ns.29.31.44.e8681

West, R. M. (2021, February 9). Best practice in statistics: Use the Welch t-test when testing the difference between two groups. *Annals of Clinical Biochemistry, 58*(4), 267-269. doi:10.1177/0004563221992088

World Health Organization. (n.d.). *Ensuring ethical standards and procedures for research with human beings*. Retrieved July 21, 2024, from World Health Organization: https://www.who.int/activities/ensuring-ethical-standards-and-procedures-for-research-with-human-beings

Xu, M., Fralick, D., Zheng, J. Z., Wang, B., Tu, X. M., & Feng, C. (2017, June 25). The Differences and Similarities Between Two-Sample T-Test and Paired T-Test. *Shanghai Arch Psychiatry, 29*(3), 184–188. doi:10.11919/j.issn.1002-0829.217070

Yip, C., Han, N.-L. R., & Sng, B. L. (2015, September). Legal and ethical issues in research. *Indian J Anaesth, 60*(9), 684-688. doi:10.4103/0019-5049.190627

GLOSSARY

Accuracy: In a classification model, it is the proportion of correct predictions out of all predictions.

Confusion Matrix: A tool used to evaluate the performance of a classification model by displaying the distribution of true positives, true negatives, false positives, and false negatives.

F1 Score: A metric that balances precision and recall by calculating their harmonic mean.

Hyperparameter tuning: Process of finding the best values for model parameters to minimise prediction error and enhance performance.

IgA: Immunoglobulin A is an antibody that plays a role in the immune system, primarily located in mucous membranes, particularly within the respiratory and digestive systems.

IgA (g/l): Immunoglobulin A levels, measured in grams per litre. The usual reference range is 0.8 to 4.0 g/l, according to UHW.

IgD: Immunoglobulin D, a class of antibodies present as an antigen receptor on most cell surfaces and predominantly found on human B cells.

IgE: Immunoglobulin E, a class of antibodies most abundant in tissue spaces, involved in the expulsion of intestinal parasites and in allergic reactions by triggering the release of histamines and leukotrienes in response to specific foreign antigens.

IgG: Immunoglobulin G is the most prevalent antibody found in blood and bodily fluids, protecting against bacterial and viral infections.

IgG (g/l): Immunoglobulin G levels, measured in grams per litre. The typical reference range is 6.0 to 16.0 g/l, as provided by UHW.

IgM: Immunoglobulin M is primarily found in blood and lymph fluid and is the first antibody produced by the body in response to a new infection. It is the largest of the antibody isotypes produced by vertebrates.

IgM (g/l): Immunoglobulin M levels, measured in grams per litre. The standard reference range is 0.5 to 2.0 g/l, as indicated by UHW.

KLC (mg/l): Kappa Light Chains levels, measured in milligrams per litre. The normal range is 3.3 to 9.4 mg/l, based on UHW guidelines.

LLC (mg/l): Lambda Light Chains levels, measured in milligrams per litre. The typical reference range is 5.70 to 26.30 mg/l, according to UHW.

NaN: In computing, it stands for "Not a Number", that is an undefined numeric value such as 0/0.

Neutrophils: A type of white blood cell that plays a crucial role in the immune system, helping the body combat infections. They are among the first immune cells to react when microorganisms like bacteria or viruses invade the body. The standard reference range is 2.5 to 7.0 thousand units per microliter of blood, as indicated by UHW.

One-hot encoding: Process that converts categorical data into a binary format, where each category is represented by a separate column with a value of 1 indicating the presence of that category and 0 indicating its absence.

PP Level: Level of paraprotein detected, with a normal value being 0 (undetectable), as defined by UHW.

PP Type: Type of paraprotein present.

Precision: In a classification model, it shows the proportion of true positives out of all instances classified as positive by the model.

Recall (*Sensitivity*): In a classification model, it reflects the proportion of true positives to all actual positive cases.

Regular expression: Sequence of symbols and characters that defines a pattern to search for in a text.

Replicability: The capacity for other researchers to achieve similar results when conducting independent studies under comparable conditions, addressing the same research question.

Reproducibility: The ability to obtain the same results using the same data, code, and analysis methods, provided these resources are accessible to others for verification.

Train/test split: Machine learning technique that consists of dividing a dataset into two subsets: a training set (for model training) and a test set (for model evaluation).

Vertical jitter: Slight random noise added to vertical positions.

Appendix A – Comorbidities Categorisation

This appendix shows the grouping of all the comorbidities in the dataset into broader categories. Each condition is classified into a specific category, such as "Cardiovascular Diseases" or "Diabetes", to simplify data analysis and improve clarity.[32]

Table 7 *Comorbidities Categorisation*

Condition	Category
atopy	Allergy
rheumatoid arthritis	Autoimmune
vasculitis	Autoimmune
lumpy breasts	Breast Disorders
thyroid cancer	Cancer, Endocrine and Metabolic Disorders
malig neop of kidney and other unspecified urinary organs	Cancer, Urological and Reproductive Disorders
malignant neoplasm of cervix uteri	Cancer, Urological and Reproductive Disorders
testicular cancer	Cancer, Urological and Reproductive Disorders
acute non-st segment elevation myocardial	Cardiovascular Diseases
acute non-st segment elevation myocardial infarction	Cardiovascular Diseases
atrial fibrillation	Cardiovascular Diseases
atrial fibrillation and flutter	Cardiovascular Diseases
benign prostatic hypertrophycongestive cardiac failure	Cardiovascular Diseases
deep vein thrombosis	Cardiovascular Diseases
deep vein thrombosis	Cardiovascular Diseases
diastolic dysfunction	Cardiovascular Diseases
essential hypertension	Cardiovascular Diseases
essential hypertension atrial fibrillation	Cardiovascular Diseases
heart failure	Cardiovascular Diseases
htn	Cardiovascular Diseases

[32] See Section 5.2.5 *Comorbidities*.

Condition	Category
hypertension	Cardiovascular Diseases
hypertensive disease	Cardiovascular Diseases
infarction	Cardiovascular Diseases
ischaemic heart disease	Cardiovascular Diseases
lvf	Cardiovascular Diseases
myocardial infarction	Cardiovascular Diseases
diabetes	Diabetes
diabetes type 2	Diabetes
dm	Diabetes
pre-diabetes	Diabetes
t2dm	Diabetes
type 2 diabetes	Diabetes
type 2 diabetes mellitus	Diabetes
chronic urinary infection	Digestive Disorders
chronic urinary tract infections	Digestive Disorders
diverticular disease	Digestive Disorders
diverticulitis	Digestive Disorders
diverticulosis	Digestive Disorders
dyspepsia	Digestive Disorders
gastro-oesophageal reflux	Digestive Disorders
irritable bowel syndrome	Digestive Disorders
irritable bowel syndrome osteoporosis	Digestive Disorders, Musculoskeletal Disorders
meniere's disease	Ear Disorders
acquired hypothyroidism	Endocrine and Metabolic Disorders
diabetes insipidus	Endocrine and Metabolic Disorders
diabetic retinopathy	Endocrine and Metabolic Disorders
gynaecomastia	Endocrine and Metabolic Disorders
hyperlipidaemia nos	Endocrine and Metabolic Disorders
hyperparathyroidism	Endocrine and Metabolic Disorders
hypothyroidism	Endocrine and Metabolic Disorders
mixed hyperlipidaemia	Endocrine and Metabolic Disorders

Condition	Category
subclinical hyperthyroidism	Endocrine and Metabolic Disorders
subclinical hypothyroidism	Endocrine and Metabolic Disorders
thyrotoxicosis	Endocrine and Metabolic Disorders
vitamin d insufficiency	Endocrine and Metabolic Disorders
mixed hyperlipidaemia type 2 diabetes mellitus	Endocrine and Metabolic Disorders, Diabetes
cataract	Eye Disorders
glaucoma	Eye Disorders
cataract cervical spondylosis	Eye Disorders, Musculoskeletal Disorders
essential (haemorrhagic) thrombocythaemia	Haematological Disorders
itp	Haematological Disorders
monocytosis	Haematological Disorders
chronic kidney disease stage 3	Kidney Disorders
injury to kidney	Kidney Disorders
kidney failure	Kidney Disorders
infective hepatitis	Liver Disorders
anxiety with depression	Mental Health Disorders
depression	Mental Health Disorders
depression nos	Mental Health Disorders
depressive disorder	Mental Health Disorders
panic disorder	Mental Health Disorders
arthritis	Musculoskeletal Disorders
arthropathy nos	Musculoskeletal Disorders
fracture nos	Musculoskeletal Disorders
gout	Musculoskeletal Disorders
knee osteoarthritis nos	Musculoskeletal Disorders
low back	Musculoskeletal Disorders
low back pain	Musculoskeletal Disorders
multiple previous fractures spine	Musculoskeletal Disorders
osteoarthritis	Musculoskeletal Disorders
osteoporosis	Musculoskeletal Disorders

Condition	Category
pain	Musculoskeletal Disorders
cva unspecified	Neurological Disorders
epilepsy	Neurological Disorders
migraine	Neurological Disorders
previous stroke	Neurological Disorders
sciatica	Neurological Disorders
shingles	Neurological Disorders
asthma	Respiratory Diseases
bronchiectasis	Respiratory Diseases
bronchitis	Respiratory Diseases
chronic obstructive airways disease	Respiratory Diseases
copd	Respiratory Diseases
pulmonary embolism	Respiratory Diseases
pulmonary embolism gout	Respiratory Diseases, Musculoskeletal Disorders
acne vulgaris	Skin Disorders
actinic keratosis	Skin Disorders
atopic dermatitis/eczema	Skin Disorders
basal cell carcinoma	Skin Disorders
basal cell carcinoma of skin	Skin Disorders
eczema	Skin Disorders
psoriasis	Skin Disorders
pyogenic granuloma	Skin Disorders
seborrhoeic eczema	Skin Disorders
squamous cell carcinoma of skin	Skin Disorders
alcohol excess	Substance Abuse Disorders
bph	Urological and Reproductive Disorders
epididymal cyst	Urological and Reproductive Disorders
erectile dysfunction	Urological and Reproductive Disorders
uterine leiomyoma - fibroids	Urological and Reproductive Disorders

Condition	Category
uterine prolapse	Urological and Reproductive Disorders
varicocele	Urological and Reproductive Disorders
prior testicular carcinoma and papillary thyroid tumour	Urological and Reproductive Disorders, Endocrine and Metabolic Disorders

Appendix B – Python Setup

Before executing any Python code, it is crucial to load a selection of essential libraries.[33] The following code snippet imports all the necessary libraries for the analysis presented in the subsequent appendices, ensuring the environment is properly configured from the outset.

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
import seaborn as sns
from geopy.geocoders import Nominatim
from geopy.distance import geodesic
from scipy.stats import ttest_ind
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix, f1_score, make_scorer, precision_score, recall_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
```

[33] See Section 3.4 *Tools and Packages Used*.

Appendix C – Dataset Importation

To integrate data from the "`Myeloma Patient Admissions UHW 1-10-23 - 30-4-24.version 20-8-24.xlsx`" Excel file[34] into a unified dataset, the following steps were executed using Python and pandas:

- File Path Definition: The path to the Excel file was defined, specifying its location on the local system.
- Reading the Admitted Patients Sheet: The sheet named "Admitted Patients" was read into a DataFrame. The header was specified to span the first two rows to accommodate multi-level headers.
- Handling Merged Cells: Since the Excel sheet contained merged cells, the script filled forward the values from the first row of headers to ensure all columns had valid names.
- Creating Combined Headers: New column headers were generated by combining the first and second row headers. This process removed any "Unnamed" placeholders and concatenated the header parts for clarity.
- Assigning New Headers: The newly created combined headers were assigned to the DataFrame, ensuring each column had a clear and distinct name.
- Reading the Non-Admitted Patients Sheet: The "Non-Admitted Patients" sheet was read into a separate DataFrame with a single row header.

After executing these steps, two DataFrames are created: one for admitted patients with appropriately combined headers and another for non-admitted patients. These DataFrames can now be used for analysis, facilitating a streamlined process compared to re-executing the import steps.

```
file_path = r"C:\Users\Victoria\Desktop\MSc DSA\Myeloma Patient Admissions
UHW 1-10-23 - 30-4-24.version 20-8-24.xlsx"

df_hosp = pd.read_excel(file_path,
                        sheet_name='Admitted Patients',
                        header=[0, 1])

first_row_filled = df_hosp.columns.get_level_values(0).tolist()

for i in range(len(first_row_filled)):
    if pd.isna(first_row_filled[i]):
        first_row_filled[i] = first_row_filled[i-1]
```

[34] See Chapter 4 *Obtain*.

```
combined_columns = []

for i in range(len(df_hosp.columns)):
    part1 = first_row_filled[i] if 'Unnamed' not in first_row_filled[i]
else ''
    part2 = df_hosp.columns.get_level_values(1)[i] if 'Unnamed' not in
df_hosp.columns.get_level_values(1)[i] else ''
    combined_columns.append(f"{part1} {part2}".strip())

df_hosp.columns = combined_columns

df_not = pd.read_excel(file_path,
                       sheet_name='Non-Admitted Patients',
                       header=0)
```

Appendix D – Dataset Concatenation

This appendix presents the code used to combine and preprocess the "Admitted Patients" and "Non-Admitted Patients" datasets into a single unified DataFrame. The following steps were executed using Python and pandas.[35]

Identify common and unique columns in both datasets.

```
columns_hosp = set(df_hosp.columns)
columns_not = set(df_not.columns)

common_columns = columns_hosp.intersection(columns_not)

specific_columns_hosp = columns_hosp - columns_not
specific_columns_not = columns_not - columns_hosp

print("Common columns:")
print(common_columns)

print("\nColumns specific to Admitted Patients (df_hosp):")
print(specific_columns_hosp)

print("\nColumns specific to Non-Admitted Patients (df_not):")
print(specific_columns_not)
```

Preprocess *df_hosp* (Admitted Patients DataFrame): replace placeholder values in *Treatment cycle number*, create a new *Number of cycles* column, add a *Hospitalised* column, and standardise column names by removing prefixes.

```
df_hosp['Treatment cycle number'] = df_hosp['Treatment cycle
number'].replace('As above', pd.NA).fillna(method='ffill')

df_hosp['Number of cycles'] = df_hosp['Treatment cycle
number'].str.extract(r'^(\d+)').fillna(0).astype(int)

def remove_prefix(column_name, prefix="Pre admission Igs  "):
    if column_name.startswith(prefix):
        return column_name[len(prefix):]
    return column_name

df_hosp['Hospitalised'] = 1

df_hosp = df_hosp.rename(columns=lambda x: remove_prefix(x))

df_hosp = df_hosp.rename(columns={
    'Current Treatment Regime                          (Select from
List)': 'Treatment',
    'postcode': 'Postcode',
    'Reason for Admission        (Select from List)': 'Reason for
Admission',
    'PP Level': 'PP',
    'Baseline Neuts': 'Neuts',
```

[35] See Section 5.2.1 *Column Names and Dataset Unification*.

```
    'PP  Type': 'PP Type',
})
```

Preprocess *df_not* (Non-Admitted Patients DataFrame): add a *Hospitalised* column, adjust column names to match *df_hosp*, and drop unnecessary columns.

```
df_not['Number'] += 100
df_not['Hospitalised'] = 0

df_not = df_not.rename(columns={
    'Number': 'Patient Number',
    'Comorbidities': 'Co-morbidities',
    'Line of therapy': 'Line of treatment',
    'Number of cycles (30th April 24)': 'Number of cycles',
    'PP (30/4/24)': 'PP',
    'Isotype': 'PP Type',
    'IgG': 'IgG (g/L)',
    'IgA': 'IgA (g/L)',
    'IgM': 'IgM (g/L)',
    'KLC': 'KLC (mg/L)',
    'LLC': 'LLC (mg/L)'
})

df_not = df_not.drop(columns=['Number.1', 'Diagnosis', 'Alive'])
```

Concatenate datasets.

```
df = pd.concat([df_hosp, df_not], ignore_index=True)
```

Appendix E – Data Cleaning

This appendix contains the code for the data-cleaning process applied to the combined dataset, which focuses on handling missing values, standardising formats, and creating derived features.

Fill missing values in columns using data from the corresponding "Post Admission/Post Discharge Igs" columns.[36]

```
column_pairs = [
    ('PP Type', 'Post admission/post discharge  Igs PP Type'),
    ('IgG (g/L)', 'Post admission/post discharge  Igs IgG'),
    ('IgA (g/L)', 'Post admission/post discharge  Igs IgA'),
    ('IgM (g/L)', 'Post admission/post discharge  Igs IgM'),
    ('KLC (mg/L)', 'Post admission/post discharge  Igs KLC'),
    ('LLC (mg/L)', 'Post admission/post discharge  Igs LLC')
]

def process_columns(df, column_pairs):
    for pre_col, post_col in column_pairs:

        df[pre_col] = df.apply(lambda row: row[post_col] if
pd.isna(row[pre_col]) else row[pre_col],axis=1)

        patient_values = df.groupby('Patient Number')[pre_col].apply(lambda
x: x.dropna().unique()).to_dict()

        df[pre_col] = df.apply(
            lambda row: patient_values.get(row['Patient Number'],
[np.nan])[0]
            if pd.isna(row[pre_col]) and
len(patient_values.get(row['Patient Number'], [])) > 0
            else row[pre_col],
            axis=1)

        df[pre_col] = df[pre_col].replace('[]', np.nan)

        if df[pre_col].dtype == 'object':
            df[pre_col] = df[pre_col].astype(str).str.strip()

    return df

df = process_columns(df, column_pairs)
```

Convert columns to numeric.

```
df['KLC (mg/L)'] = pd.to_numeric(df['KLC (mg/L)'], errors='coerce')
df['LLC (mg/L)'] = pd.to_numeric(df['LLC (mg/L)'], errors='coerce')
```

Calculate *Ratio* column.[37]

```
df['Ratio'] = df['KLC (mg/L)'] / df['LLC (mg/L)']
```

[36] See Section 5.2.2 *Missing Values for Pre- and Post-Admission Data*.
[37] See Section 5.2.3 *Ratio Calculation (KLC/LLC)*.

```
df['Ratio'].replace([np.inf, -np.inf], np.nan, inplace=True)
```

Parse and standardise date formats and calculate *Age*.[38]

```
def parse_date(date_str):
    if isinstance(date_str, pd.Timestamp):
        return date_str.date()
    elif isinstance(date_str, str):
        date_str = date_str.strip()
        try:
            return pd.to_datetime(date_str, errors='coerce').date()
        except ValueError:
            return pd.NaT
    return pd.NaT

df['DOB'] = df['DOB'].apply(parse_date).pipe(pd.to_datetime,
errors='coerce').dt.normalize()
df['Date of Admission'] = pd.to_datetime(df['Date of Admission'],
errors='coerce')

default_date = pd.Timestamp('2024-04-30')
df['Age'] = df.apply(
    lambda row: ((default_date if pd.isna(row['Date of Admission']) else
row['Date of Admission']) - row['DOB']).days // 365, axis=1)
```

Process *Co-morbidities* column: fill NaN values with an empty string, define mappings from conditions to categories, apply mapping of conditions to categories, manually fix any remaining issues, and create one column per category with a unit if the category is present in the patient and a 0 if it is not.[39]

```
df['Co-morbidities'] = df['Co-morbidities'].fillna('')

condition_to_category = {
    'atopy': 'Allergy',
    'rheumatoid arthritis': 'Autoimmune',
    'vasculitis': 'Autoimmune',
    'acute non-st segment elevation myocardial': 'Cardiovascular Diseases',
    'acute non-st segment elevation myocardial infarction': 'Cardiovascular
Diseases',
    'atrial fibrillation': 'Cardiovascular Diseases',
    'deep vein thrombosis': 'Cardiovascular Diseases',
    'diastolic dysfunction': 'Cardiovascular Diseases',
    'essential hypertension': 'Cardiovascular Diseases',
    'essential hypertension atrial fibrillation': 'Cardiovascular
Diseases',
    'heart failure': 'Cardiovascular Diseases',
    'htn': 'Cardiovascular Diseases',
    'hypertension': 'Cardiovascular Diseases',
    'infarction': 'Cardiovascular Diseases',
    'ischaemic heart disease': 'Cardiovascular Diseases',
    'lvf': 'Cardiovascular Diseases',
    'myocardial infarction': 'Cardiovascular Diseases',
    'atrial fibrillation and flutter': 'Cardiovascular Diseases',
```

[38] See Section 5.2.4 *Date Formats*.
[39] See Section 5.2.5 *Comorbidities*.

```
    'benign prostatic hypertrophycongestive cardiac failure':
'Cardiovascular Diseases',
    'hypertensive disease': 'Cardiovascular Diseases',
    'deep vein thrombosis': 'Cardiovascular Diseases',
    'depression': 'Mental Health Disorders',
    'depression nos': 'Mental Health Disorders',
    'depressive disorder': 'Mental Health Disorders',
    'anxiety with depression': 'Mental Health Disorders',
    'panic disorder': 'Mental Health Disorders',
    'acquired hypothyroidism': 'Endocrine and Metabolic Disorders',
    'diabetic retinopathy': 'Endocrine and Metabolic Disorders',
    'hyperlipidaemia nos': 'Endocrine and Metabolic Disorders',
    'hypothyroidism': 'Endocrine and Metabolic Disorders',
    'mixed hyperlipidaemia': 'Endocrine and Metabolic Disorders',
    'subclinical hypothyroidism': 'Endocrine and Metabolic Disorders',
    'gynaecomastia': 'Endocrine and Metabolic Disorders',
    'hyperparathyroidism': 'Endocrine and Metabolic Disorders',
    'thyroid cancer': 'Cancer, Endocrine and Metabolic Disorders',
    'thyrotoxicosis': 'Endocrine and Metabolic Disorders',
    'vitamin d insufficiency': 'Endocrine and Metabolic Disorders',
    'diabetes insipidus': 'Endocrine and Metabolic Disorders',
    'subclinical hyperthyroidism': 'Endocrine and Metabolic Disorders',
    'diabetes': 'Diabetes',
    'diabetes type 2': 'Diabetes',
    'dm': 'Diabetes',
    't2dm': 'Diabetes',
    'type 2 diabetes': 'Diabetes',
    'type 2 diabetes mellitus': 'Diabetes',
    'pre-diabetes': 'Diabetes',
    'mixed hyperlipidaemia type 2 diabetes mellitus': 'Endocrine and
Metabolic Disorders, Diabetes',
    'arthritis': 'Musculoskeletal Disorders',
    'arthropathy nos': 'Musculoskeletal Disorders',
    'fracture nos': 'Musculoskeletal Disorders',
    'gout': 'Musculoskeletal Disorders',
    'knee osteoarthritis nos': 'Musculoskeletal Disorders',
    'low back': 'Musculoskeletal Disorders',
    'low back pain': 'Musculoskeletal Disorders',
    'multiple previous fractures spine': 'Musculoskeletal Disorders',
    'osteoarthritis': 'Musculoskeletal Disorders',
    'osteoporosis': 'Musculoskeletal Disorders',
    'pain': 'Musculoskeletal Disorders',
    'cataract cervical spondylosis': 'Eye Disorders, Musculoskeletal
Disorders',
    'irritable bowel syndrome osteoporosis': 'Digestive Disorders,
Musculoskeletal Disorders',
    'psoriasis': 'Skin Disorders',
    'actinic keratosis': 'Skin Disorders',
    'acne vulgaris': 'Skin Disorders',
    'atopic dermatitis/eczema': 'Skin Disorders',
    'basal cell carcinoma': 'Skin Disorders',
    'basal cell carcinoma of skin': 'Skin Disorders',
    'eczema': 'Skin Disorders',
    'pyogenic granuloma': 'Skin Disorders',
    'seborrhoeic eczema': 'Skin Disorders',
    'squamous cell carcinoma of skin': 'Skin Disorders',
    'asthma': 'Respiratory Diseases',
    'bronchiectasis': 'Respiratory Diseases',
    'bronchitis': 'Respiratory Diseases',
```

```
    'chronic obstructive airways disease': 'Respiratory Diseases',
    'copd': 'Respiratory Diseases',
    'pulmonary embolism': 'Respiratory Diseases',
    'pulmonary embolism gout': 'Respiratory Diseases, Musculoskeletal
Disorders',
    'cva unspecified': 'Neurological Disorders',
    'epilepsy': 'Neurological Disorders',
    'migraine': 'Neurological Disorders',
    'previous stroke': 'Neurological Disorders',
    'sciatica': 'Neurological Disorders',
    'shingles': 'Neurological Disorders',
    'meniere\'s disease': 'Ear Disorders',
    'glaucoma': 'Eye Disorders',
    'cataract': 'Eye Disorders',
    'chronic urinary infection': 'Digestive Disorders',
    'chronic urinary tract infections': 'Digestive Disorders',
    'diverticulitis': 'Digestive Disorders',
    'diverticulosis': 'Digestive Disorders',
    'diverticular disease': 'Digestive Disorders',
    'gastro-oesophageal reflux': 'Digestive Disorders',
    'irritable bowel syndrome': 'Digestive Disorders',
    'dyspepsia': 'Digestive Disorders',
    'infective hepatitis': 'Liver Disorders',
    'itp': 'Haematological Disorders',
    'monocytosis': 'Haematological Disorders',
    'essential (haemorrhagic) thrombocythaemia': 'Haematological
Disorders',
    'chronic kidney disease stage 3': 'Kidney Disorders',
    'injury to kidney': 'Kidney Disorders',
    'kidney failure': 'Kidney Disorders',
    'lumpy breasts': 'Breast Disorders',
    'alcohol excess': 'Substance Abuse Disorders',
    'erectile dysfunction': 'Urological and Reproductive Disorders',
    'epididymal cyst': 'Urological and Reproductive Disorders',
    'bph': 'Urological and Reproductive Disorders',
    'malig neop of kidney and other unspecified urinary organs': 'Cancer,
Urological and Reproductive Disorders',
    'prior testicular carcinoma and papillary thyroid tumour': 'Urological
and Reproductive Disorders, Endocrine and Metabolic Disorders',
    'malignant neoplasm of cervix uteri': 'Cancer, Urological and
Reproductive Disorders',
    'testicular cancer': 'Cancer, Urological and Reproductive Disorders',
    'uterine leiomyoma - fibroids': 'Urological and Reproductive
Disorders',
    'uterine prolapse': 'Urological and Reproductive Disorders',
    'varicocele': 'Urological and Reproductive Disorders'
}

def map_to_category(condition):
    categories = []

    conditions = re.split(r'\n+', condition)
    conditions = [re.sub(r'[.\s]+', ' ', c).strip().lower() for c in
conditions if c.strip()]

    for cond in conditions:
        sub_conditions = [sub_cond.strip(',').strip() for sub_cond in
re.split(r',\s*', cond) if sub_cond.strip()]
        for sub_cond in sub_conditions:
```

```
            if sub_cond in ['dm', 't2dm', 'type 2 diabetes mellitus',
'diabetes type 2']:
                categories.append('Diabetes')
            elif sub_cond in ['hypertensive', 'hypertensive disease']:
                categories.append('Cardiovascular Diseases')
            else:
                categories.append(condition_to_category.get(sub_cond,
'Other'))

    return categories

df['Category'] = df['Co-morbidities'].apply(map_to_category)

def fix_combined_categories(categories):
    fixed_categories = []
    for category in categories:
        if category == 'Endocrine and Metabolic Disorders, Diabetes':
            fixed_categories.extend(['Endocrine and Metabolic Disorders',
'Diabetes'])
        elif category == 'Digestive Disorders, Musculoskeletal Disorders':
            fixed_categories.extend(['Digestive Disorders',
'Musculoskeletal Disorders'])
        elif category == 'Cancer, Endocrine and Metabolic Disorders':
            fixed_categories.extend(['Cancer', 'Endocrine and Metabolic
Disorders'])
        elif category == 'Cancer, Urological and Reproductive Disorders':
            fixed_categories.extend(['Cancer', 'Urological and Reproductive
Disorders'])
        else:
            fixed_categories.append(category)
    return fixed_categories

df['Category'] = df['Category'].apply(lambda x: [item.strip() for item in
x[0].split(',')] if isinstance(x, list) and len(x) == 1 else x)

df['Category'] = df['Category'].apply(fix_combined_categories)

category_columns = set(category for row in df['Category'] for category in
row)

for category in category_columns:
    df[category] = df['Category'].apply(lambda x: x.count(category))
```

Fix *Postcode* and calculate *Distance to UHW* (CF14 4XW). Assign the median value to the NaN values in the *Distance from UHW* column.[40]

```
postcode_regex = r'^[A-Z]{2}\d{1,2}\s\d[A-Z]{2}$'
invalid_postcodes = df[~df['Postcode'].str.match(postcode_regex, na=False)]

def fix_postcode(postcode):
    if pd.isna(postcode):
        return postcode
    postcode = postcode.strip().upper()

    if postcode.startswith('CF0'):
        postcode = 'CF' + postcode[3:]
```

[40] See Section 5.2.6 *Postcodes*.

```
    if len(postcode) == 6 and postcode[3] != ' ':
        return f"{postcode[:3]} {postcode[3:]}"
    elif len(postcode) == 7 and postcode[4] != ' ':
        return f"{postcode[:4]} {postcode[4:]}"
    elif postcode.startswith(r'\t'):
        return postcode.lstrip(r'\t')
    else:
        return postcode

df['Postcode_fixed'] = df['Postcode'].apply(fix_postcode)
geolocator = Nominatim(user_agent="postcode_geocoder")

cache = {
    'CF4  3LW': (51.498539, -3.185559),
    'CF4  9AH': (51.536119, -3.203837)}

def get_coordinates(postcode):
    postcode = postcode.strip().upper()

    if postcode in cache:
        return cache[postcode]

    if not isinstance(postcode, str) or pd.isna(postcode) or postcode ==
'':
        return (None, None)

    location = geolocator.geocode(postcode + ", UK")
    if location is None:
        coords = (None, None)
    else:
        coords = (location.latitude, location.longitude)

    cache[postcode] = coords
    return coords

def calculate_distance(postcode1, postcode2):
    coords_1 = get_coordinates(postcode1)
    coords_2 = get_coordinates(postcode2)

    if None in coords_1 or None in coords_2:
        return None

    return geodesic(coords_1, coords_2).kilometers

reference_postcode = 'CF14 4XW'

df['Latitude'] = df['Postcode_fixed'].apply(lambda x: get_coordinates(x)[0]
if isinstance(x, str) and x.strip() != '' else None)
df['Longitude'] = df['Postcode_fixed'].apply(lambda x:
get_coordinates(x)[1] if isinstance(x, str) and x.strip() != '' else None)
df['Distance from UHW'] = df.apply(lambda row:
calculate_distance(row['Postcode_fixed'], reference_postcode) if
pd.notna(row['Latitude']) and pd.notna(row['Longitude']) else None, axis=1)

median_distance = df['Distance from UHW'].median()

df['Distance from UHW'].fillna(median_distance, inplace=True)
```

Preprocess *PP Type* column.[41]

```
df['PP Type Copy'] = df['PP Type']

df['PP Type Copy'] = df['PP Type Copy'].astype(str)
df['PP Type Copy'] = df['PP Type Copy'].fillna('')
df['PP Type Copy'] = df['PP Type Copy'].replace('IgG Kappa\nIgG lambda\nIgM
kappa\n on IF', 'IgG\nIgG\nIgM')
df['PP Type Copy'] = df['PP Type Copy'].str.replace(' band 1', '',
regex=False)
df['PP Type Copy'] = df['PP Type Copy'].str.replace(' band 2', '',
regex=False)
df['PP Type Copy'] = df['PP Type Copy'].str.replace(', ,', ',',
regex=False)
df['PP Type Copy'] = df['PP Type Copy'].str.strip()
df['PP Type Copy'] = df['PP Type Copy'].str.lower()

replacements = {
    'fllc only': 'fllc',
    'igm kappa on if': 'igm',
    'igg kappa on if': 'igg',
    'igg lambda on if': 'igg',
    'free lambda on if': 'free',
    'igm kappa band': 'igm',
    'light chain (k)': 'light chain',
    'lambda lc 277 pp in progress': 'lambda',
    'igg lambda and kappa': ['igg', 'igg'],
    'igg kappa\nigg lambda\n on if' : ['igg', 'igg'],
    ' kappa': '',
    ' lambda': '',
    'free': 'light chain',
    'fllc': 'light chain',
    'lambda': 'light chain',
    'kappa': 'light chain',
    'nan': ''
}

for old, new in replacements.items():
    if isinstance(new, str):
        df['PP Type Copy'] = df['PP Type Copy'].str.replace(old, new,
case=False, regex=False)
    elif isinstance(new, list):
        df['PP Type Copy'] = df['PP Type Copy'].str.replace(old,
'\n'.join(new), case=False, regex=False)


df['PP Type Copy'] = df['PP Type Copy'].str.split('\n')
df[['PP Type','PP Type Copy']]
df_flat = df.explode('PP Type Copy')
one_hot = pd.get_dummies(df_flat['PP Type Copy']).astype(int)

df_encoded = df_flat.join(one_hot)
df_encoded =
df_encoded.groupby(df_encoded.index).max().reset_index(drop=True)

existing_columns = set(df.columns)
```

[41] See Section 5.2.7 *PP Type*.

```
new_columns_to_keep = [col for col in df_encoded.columns if col not in
existing_columns]

df = pd.concat([df, df_encoded[new_columns_to_keep]], axis=1)
df = df.loc[:, df.columns != '']
```

Preprocess *Number of Cycles* column.[42]

```
df['Number of cycles'] = df['Number of cycles'].astype(str)
df['Number of cycles'] = df['Number of cycles'].str.replace('th', '',
regex=False)
```

Preprocess *Line of Treatment* column.[43]

```
df['Line of treatment'] = df['Line of treatment'].astype(str)
df['Line of treatment'] = df['Line of treatment'].str.replace('nd', '',
regex=False)
df['Line of treatment'] = df['Line of treatment'].str.replace('th', '',
regex=False)
df['Line of treatment'] = df['Line of treatment'].str.replace('
previously', '', regex=False)
df['Line of treatment'] = df['Line of treatment'].str.replace('st', '',
regex=False)
df['Line of treatment'] = df['Line of treatment'].str.replace('rd', '',
regex=False)

def replace_less_than(value):
    if isinstance(value, str) and value.startswith('<'):
        number = float(re.search(r'\d+\.\d+|\d+', value).group())
        result = round(0.75 * number, 2)
        return result if result != 0 else number
    return value
```

Preprocess *IgA* and *IgM* columns.[44]

```
df['IgA (g/L)'] = df['IgA (g/L)'].apply(replace_less_than)
df['IgM (g/L)'] = df['IgM (g/L)'].apply(replace_less_than)
df['IgA (g/L)'] = df['IgA (g/L)'].replace('<0.1', 0.075)
df['IgM (g/L)'] = df['IgM (g/L)'].replace('<0.1', 0.075)
```

Convert *Patient Number*, *Hospitalised*, *IgG (g/L)*, *IgA (g/L)*, *IgM (g/L)*, *Neuts*, *Number of cycles* columns to numeric.

```
cols_to_convert = ['Patient Number', 'Hospitalised', 'IgG (g/L)', 'IgA
(g/L)', 'IgM (g/L)', 'Neuts', 'Number of cycles']
df[cols_to_convert] = df[cols_to_convert].replace('nan', np.nan)
df[cols_to_convert] = df[cols_to_convert].apply(pd.to_numeric,
errors='coerce')
```

Remove duplicate patient records to create a new DataFrame with each patient represented only once.[45]

```
df_sorted = df.sort_values(by=['Patient Number', 'Date of Admission'])
```

[42] See Section 5.2.8 *Number of Cycles and Line of Treatment*.
[43] See Section 5.2.8 *Number of Cycles and Line of Treatment*.
[44] See Section 5.2.9 *IgA and IgM Values*.
[45] See Section 5.2.10 *Removing Duplicate Records*.

```
df_unique = df_sorted.drop_duplicates(subset=['Patient Number'],
keep='first')

df_unique.reset_index(drop=True, inplace=True)
```

Standardise *Gender* column.[46]

```
def standardise_gender(gender):
    if gender in ['M', 'Male']:
        return 'M'
    elif gender in ['F', 'Female']:
        return 'F'
    else:
        return gender

df_unique['Gender'] = df_unique['Gender'].apply(standardise_gender)
```

Standardise *Ethnicity* column.[47]

```
def standardise_ethnicity(ethnicity):
    if ethnicity in ['Caucasian (Italian)', 'Caucasian/Italian',
'Caucasian/Polish']:
        return 'Caucasian Europe'
    elif ethnicity == 'Caucasian':
        return 'Caucasian UK'
    elif ethnicity in ['Indian', 'Bangladeshi', 'Sinhalese']:
        return 'South Asian'
    elif ethnicity in ['Sudanese', 'African', 'Afrocarribbean']:
        return 'African'
    else:
        return ethnicity

df_unique['Ethnicity'] =
df_unique['Ethnicity'].apply(standardise_ethnicity)
```

Identify and fix missing values (replace with the mean for each column).[48]

```
missing_values = df.isna().sum()

filtered_missing_values = missing_values[(missing_values > 0) &
(missing_values < 95)]
filtered_missing_values =
filtered_missing_values.drop(['Postcode','Postcode_fixed','Latitude','Longi
tude','DOB'], errors='ignore')

print("Missing values:")
print(filtered_missing_values)

columns_to_fill = ['IgG (g/L)', 'KLC (mg/L)', 'LLC (mg/L)', 'IgA (g/L)',
'IgM (g/L)', 'PP', 'Neuts', 'Number of cycles', 'Ratio', 'Age']

for column in columns_to_fill:
    df[column] = pd.to_numeric(df[column], errors='coerce')
```

[46] See Section 5.2.11 *Gender*.
[47] See Section 5.2.12 *Ethnicity*.
[48] See Section 5.2.13 *Handling Residual Missing Values*.

```
for column in columns_to_fill:
    df[column].fillna(df[column].mean(), inplace=True)
```

Remove columns that are not common to both original datasets and those that have already been used to create derived columns. Also, drop the *Treatment* column due to its high variability with too few cases.[49]

```
df_unique = df_unique.dropna(thresh=df_unique.shape[0] - 95 + 1, axis=1)

df_unique = df_unique.loc[:, df_unique.apply(lambda col: (col == '').sum()
< 95)]

columns_to_remove = [
    'Postcode', 'Postcode_fixed', 'Latitude', 'Longitude',
    'PP Type', 'Co-morbidities', 'DOB', 'Category', 'PP Type Copy', 'PP',
'Treatment']

df_unique = df_unique.drop(columns=columns_to_remove)
```

[49] See Section 5.2.14 *Final Data Cleaning and Column Removal*.

Appendix F – Descriptive Analysis

The code below was utilised to conduct the descriptive analysis of the dataset.

Generate summary statistics and data about categorical data.[50]

```
df_unique.describe()

categorical_columns = ['Gender', 'Ethnicity']

print("\nCategorical Data Distribution:")
for col in categorical_columns:
    print(f'\n{col}:')
    print(df_unique[col].value_counts())
```

Split data in *Hospitalised* and *Not Hospitalised* groups for histograms.

```
group1 = df_unique[df_unique['Hospitalised'] == 1]
group2 = df_unique[df_unique['Hospitalised'] == 0]
```

Create *Age* histograms.[51]

```
min_age = int(df_unique['Age'].min())
max_age = int(df_unique['Age'].max()) + 4
bins = list(range(min_age, max_age, 3))

age_data_group1 = group1['Age'].dropna()
age_data_group2 = group2['Age'].dropna()

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
counts_group1, edges_group1, patches_group1 = plt.hist(age_data_group1,
bins=bins, edgecolor='black', color='salmon')
x_tick_labels_group1 = [f'{edges_group1[i]:.0f}-{edges_group1[i+1] -
1:.0f}' for i in range(len(edges_group1) - 1)]
midpoints_group1 = [(edges_group1[i] + edges_group1[i+1]) / 2 for i in
range(len(edges_group1) - 1)]
plt.xticks(ticks=midpoints_group1, labels=x_tick_labels_group1,
rotation=45, fontsize=12)
plt.yticks(fontsize=12)
plt.title('Histogram of Age (3-Year Intervals) - Hospitalised',
fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(True)

plt.subplot(1, 2, 2)
counts_group2, edges_group2, patches_group2 = plt.hist(age_data_group2,
bins=bins, edgecolor='black', color='skyblue')
x_tick_labels_group2 = [f'{edges_group2[i]:.0f}-{edges_group2[i+1] -
1:.0f}' for i in range(len(edges_group2) - 1)]
midpoints_group2 = [(edges_group2[i] + edges_group2[i+1]) / 2 for i in
range(len(edges_group2) - 1)]
```

[50] See Section 6.1 *Summary Statistics*.
[51] See Section 6.2 *Age Distribution by Hospitalisation Status*.

```
plt.xticks(ticks=midpoints_group2, labels=x_tick_labels_group2,
rotation=45, fontsize=12)
plt.yticks(fontsize=12)
plt.title('Histogram of Age (3-Year Intervals) - Not Hospitalised',
fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()
```

Plot *Line of Treatment* histograms.[52]

```
line_of_treatment_counts_group1 = group1['Line of
treatment'].value_counts()
line_of_treatment_counts_group2 = group2['Line of
treatment'].value_counts()

all_treatment_lines = sorted(set(line_of_treatment_counts_group1.index) |
set(line_of_treatment_counts_group2.index))

line_of_treatment_counts_group1 =
line_of_treatment_counts_group1.reindex(all_treatment_lines, fill_value=0)
line_of_treatment_counts_group2 =
line_of_treatment_counts_group2.reindex(all_treatment_lines, fill_value=0)

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.bar(line_of_treatment_counts_group1.index,
line_of_treatment_counts_group1.values, color='salmon', edgecolor='black')
plt.title('Line of Treatment Distribution - Hospitalised', fontsize=14)
plt.xlabel('Line of Treatment', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xticks(rotation=0, ha='right', fontsize=12)
plt.yticks(fontsize=12)
plt.grid(True)

plt.subplot(1, 2, 2)
plt.bar(line_of_treatment_counts_group2.index,
line_of_treatment_counts_group2.values, color='skyblue', edgecolor='black')
plt.title('Line of Treatment Distribution - Not Hospitalised', fontsize=14)
plt.xlabel('Line of Treatment', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.xticks(rotation=0, ha='right', fontsize=12)
plt.yticks(fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()
```

Draw *Number of Cycles* histogram.[53]

```
min_cycles = int(df_unique['Number of cycles'].min())
max_cycles = int(df_unique['Number of cycles'].max()) + 4
```

[52] See Section 6.3 *Line of Treatment Distribution by Hospitalisation Status*.
[53] See Section 6.4 *Number of Cycles Distribution by Hospitalisation Status*.

```
bins = list(range(min_cycles, max_cycles, 3))

cycles_data_group1 = group1['Number of cycles'].dropna()
cycles_data_group2 = group2['Number of cycles'].dropna()

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
counts_group1, edges_group1, patches_group1 = plt.hist(cycles_data_group1,
bins=bins, edgecolor='black', color='salmon')
x_tick_labels_group1 = [f'{edges_group1[i]:.0f}-{edges_group1[i+1] -
1:.0f}' for i in range(len(edges_group1) - 1)]
midpoints_group1 = [(edges_group1[i] + edges_group1[i+1]) / 2 for i in
range(len(edges_group1) - 1)]
plt.xticks(ticks=midpoints_group1, labels=x_tick_labels_group1,
rotation=60, fontsize=12)
plt.yticks(fontsize=12)
plt.title('Histogram of Number of Cycles - Hospitalised', fontsize=14)
plt.xlabel('Number of Cycles', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(True)

plt.subplot(1, 2, 2)
counts_group2, edges_group2, patches_group2 = plt.hist(cycles_data_group2,
bins=bins, edgecolor='black', color='skyblue')
x_tick_labels_group2 = [f'{edges_group2[i]:.0f}-{edges_group2[i+1] -
1:.0f}' for i in range(len(edges_group2) - 1)]
midpoints_group2 = [(edges_group2[i] + edges_group2[i+1]) / 2 for i in
range(len(edges_group2) - 1)]
plt.xticks(ticks=midpoints_group2, labels=x_tick_labels_group2,
rotation=60, fontsize=12)
plt.yticks(fontsize=12)
plt.title('Histogram of Number of Cycles - Not Hospitalised', fontsize=14)
plt.xlabel('Number of Cycles', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.grid(True)

plt.tight_layout()
plt.show()
```

Create scatter plot showing the *Number of Cycles* vs. *Line of Treatment* by Hospitalisation Status.[54]

```
np.random.seed(14)

df_unique['Line of treatment'] = pd.to_numeric(df_unique['Line of
treatment'], errors='coerce')

jitter = 0.25
df_unique['Number of cycles jittered'] = df_unique['Number of cycles']
df_unique['Line of treatment jittered'] = df_unique['Line of treatment'] +
np.random.uniform(-jitter, jitter, size=len(df_unique))

plt.figure(figsize=(14, 4))

sns.scatterplot(
```

[54] See Section 6.5 *Number of Cycles vs. Line of Treatment by Hospitalisation Status*.

```
    data=df_unique[df_unique['Hospitalised'] == 0],
    x='Number of cycles jittered',
    y='Line of treatment jittered',
    color='deepskyblue',
    s=100,
    marker='o',
    alpha=0.5,
    edgecolor='black',
    label='Not Hospitalised'
)

sns.scatterplot(
    data=df_unique[df_unique['Hospitalised'] == 1],
    x='Number of cycles jittered',
    y='Line of treatment jittered',
    color='tomato',
    s=80,
    marker='X',
    alpha=0.6,
    edgecolor='black',
    label='Hospitalised'
)

plt.xlabel('Number of Cycles', fontsize=12)
plt.ylabel('Line of Treatment', fontsize=12)
plt.title('Number of Cycles vs. Line of Treatment by Hospitalisation
Status', fontsize=14)

plt.grid(True, linestyle='--', alpha=0.7)

plt.xticks(ticks=np.arange(df_unique['Number of cycles'].min(),
df_unique['Number of cycles'].max() + 10, 10), fontsize=12)

plt.yticks(fontsize=12)

plt.legend(title='Hospitalisation Status', fontsize=12, title_fontsize=12)
plt.show()
```

Create plots of all the comorbidities columns.[55]

```
disorders = [
    'Skin Disorders', 'Endocrine and Metabolic Disorders',
    'Urological and Reproductive Disorders', 'Cardiovascular Diseases',
    'Autoimmune', 'Kidney Disorders', 'Digestive Disorders',
    'Haematological Disorders', 'Substance Abuse Disorders',
    'Eye Disorders', 'Respiratory Diseases', 'Liver Disorders',
    'Mental Health Disorders', 'Allergy', 'Cancer', 'Ear Disorders',
    'Diabetes', 'Neurological Disorders', 'Breast Disorders',
    'Musculoskeletal Disorders']

differences = {}
coefficients = {}
for disorder in disorders:
    hosp_percent = df_unique[df_unique['Hospitalised'] ==
1][disorder].mean() * 100
    non_hosp_percent = df_unique[df_unique['Hospitalised'] ==
0][disorder].mean() * 100
```

[55] See Section 6.6 *Health Disorders Distribution by Hospitalisation Status*.

```
    difference = hosp_percent - non_hosp_percent
    coefficient = hosp_percent / non_hosp_percent if non_hosp_percent != 0
else float('inf')
    differences[disorder] = difference
    coefficients[disorder] = coefficient

sorted_disorders = sorted(differences.keys(), key=lambda x: differences[x],
reverse=True)

fig, axes = plt.subplots(nrows=5, ncols=4, figsize=(20, 20))
axes = axes.flatten()

for i, disorder in enumerate(sorted_disorders):
    hosp_percent = df_unique[df_unique['Hospitalised'] ==
1][disorder].mean() * 100
    non_hosp_percent = df_unique[df_unique['Hospitalised'] ==
0][disorder].mean() * 100
    difference = differences[disorder]
    coefficient = coefficients[disorder]

    axes[i].bar(['Hospitalised', 'Non-Hospitalised'], [hosp_percent,
non_hosp_percent], color=['salmon', 'skyblue'])

    axes[i].set_title(f'{disorder}\nDiff: {difference:.2f}% | Coeff:
{coefficient:.2f}', fontsize=20)
    axes[i].set_ylabel('Percentage (%)', fontsize=18)
    axes[i].set_ylim(0, 100)
    axes[i].tick_params(axis='x', labelsize=18)
    axes[i].tick_params(axis='y', labelsize=18)

plt.tight_layout()

plt.show()
```

Generate correlation matrix and calculate correlation values.[56]

```
columns_to_exclude = ['Line of treatment jittered', 'Patient Number']
numeric_columns = df_unique.select_dtypes(include=['number']).columns
columns_to_select = [col for col in numeric_columns if col not in
columns_to_exclude]
selected_columns = df_unique[columns_to_select]

plt.figure(figsize=(10, 8))

correlation_matrix = selected_columns.corr()

high_corr = correlation_matrix[(correlation_matrix.abs() > 0.5) &
(correlation_matrix != 1.0)].dropna(how='all', axis=0).dropna(how='all',
axis=1)

sns.heatmap(high_corr, annot=True, fmt=".2f", cmap='coolwarm', cbar=True,
annot_kws={"size": 14},
            linewidths=1, linecolor='black')

plt.title('Correlation Matrix (|Correlation| > 0.5)', fontsize=16)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
```

[56] See Section 6.7 *Correlation Analysis*.

```
cbar = plt.gcf().axes[-1]
cbar.tick_params(labelsize=14)

plt.show()

columns_to_exclude = ['Line of treatment jittered', 'Patient Number']

corr_pairs = correlation_matrix.unstack()
sorted_corr_pairs = corr_pairs.sort_values(key=lambda x: abs(x),
ascending=False)

sorted_corr_pairs = sorted_corr_pairs[sorted_corr_pairs != 1]

sorted_corr_df = sorted_corr_pairs.reset_index()
sorted_corr_df.columns = ['Variable1', 'Variable2', 'Correlation']

seen_pairs = set()

filtered_corr_df = sorted_corr_df[sorted_corr_df.apply(lambda row:
(row['Variable1'], row['Variable2']) not in seen_pairs and  not
seen_pairs.add((row['Variable2'], row['Variable1'])), axis=1)]

top_positive_corr = filtered_corr_df[filtered_corr_df['Correlation'] >
0].head(10)
top_negative_corr = filtered_corr_df[filtered_corr_df['Correlation'] <
0].head(10)

print("Top positively correlated variable pairs:")
print(top_positive_corr)

print("\nTop negatively correlated variable pairs:")
print(top_negative_corr)
```

Appendix G – Paired T-Tests

The provided Python code served the purpose of carrying out the paired t-test analyses.

Carry out the one-tailed two-sample t-test to see if the mean *Number of Cycles* of the Hospitalised group is smaller than that of the Not Hospitalised group, interpreting with 95% and 99% confidence.[57]

```
hospitalised_group = df_unique[df_unique['Hospitalised'] == 1]['Number of
cycles']
non_hospitalised_group = df_unique[df_unique['Hospitalised'] == 0]['Number
of cycles']

hospitalised_group = hospitalised_group.dropna()
non_hospitalised_group = non_hospitalised_group.dropna()

t_stat, p_value = ttest_ind(hospitalised_group, non_hospitalised_group,
alternative='less', equal_var=False)

print(f"One-Tailed Independent Samples T-Test Results:")
print(f"T-statistic: {t_stat}")
print(f"P-value: {p_value}")

alpha = 0.05
if p_value < alpha:
    print("With 95% confidence, we reject the null hypothesis and conclude
that the mean number of cycles for the hospitalised group is significantly
smaller than that of the non-hospitalised group.")
else:
    print("We fail to reject the null hypothesis. There is no significant
evidence to suggest that the mean number of cycles for the hospitalised
group is smaller than that of the non-hospitalised group.")


alpha2 = 0.01
if p_value < alpha2:
    print("With 99% confidence, we reject the null hypothesis and conclude
that the mean number of cycles for the hospitalised group is significantly
smaller than that of the non-hospitalised group.")
else:
    print("We fail to reject the null hypothesis. There is no significant
evidence to suggest that the mean number of cycles for the hospitalised
group is smaller than that of the non-hospitalised group.")
```

Carry out the two-sided two-sample t-test to see if the mean *Number of Cycles* of the Hospitalised group is different than that of the Not Hospitalised group, interpreting with 95%.[58]

```
hospitalised_group = df_unique[df_unique['Hospitalised'] == 1]['Line of
treatment']
```

[57] See Section 7.1.1 *Comparison of Treatment Cycles by Hospitalisation Status*.
[58] See Section 7.1.2 *Comparison of Treatment Lines by Hospitalisation Status*.

```
non_hospitalised_group = df_unique[df_unique['Hospitalised'] == 0]['Line of
treatment']

hospitalised_group = hospitalised_group.dropna()
non_hospitalised_group = non_hospitalised_group.dropna()

t_stat, p_value = ttest_ind(hospitalised_group, non_hospitalised_group,
alternative='two-sided', equal_var=False)

print(f"One-Tailed Independent Samples T-Test Results:")
print(f"T-statistic: {t_stat}")
print(f"P-value: {p_value}")

alpha = 0.05
if p_value < alpha:
    print("With 95% confidence, we reject the null hypothesis and conclude
that there is a significant difference in the mean number of treatment
lines between the hospitalised and non-hospitalised groups.")
else:
    print("We fail to reject the null hypothesis. There is no significant
evidence to suggest that there is a difference in the mean number of
treatment lines between the hospitalised and non-hospitalised groups.")
```

Appendix H – Logistic Regression for Hospital Admission Risk Factors

This appendix details the Python code used to develop a logistic regression model to identify key factors contributing to hospital admission.

Build logistic regression and present classification report, confusion matrix and accuracy score calculations.[59]

```
features = df_unique.drop(columns=['Hospitalised', 'Patient Number',
'Ethnicity', 'Line of treatment jittered' ])
target = df_unique['Hospitalised']

features = pd.get_dummies(features, drop_first=True)

log_reg = LogisticRegression(max_iter=1000)
X_train, X_test, y_train, y_test = train_test_split(features, target,
test_size=0.3, random_state=14)
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_test)

feature_names = features.columns
coefficients = log_reg.coef_[0]
coef_df = pd.DataFrame({
    'Feature': feature_names,
    'Coefficient': coefficients})
coef_df['Absolute Coefficient'] = coef_df['Coefficient'].abs()
print("Coefficients shape:", coefficients.shape)

significant_coef_df = coef_df[coef_df['Absolute Coefficient'] > 0.3]
significant_features = significant_coef_df['Feature']

valid_features = [feature for feature in significant_features if feature in
features.columns]
features_filtered = features[valid_features]

def count_non_zero_or_false(column):
    if column.dtype == 'object':
        return (column != 'False').sum()
    else:
        return (column != 0).sum()

row_counts = features_filtered.apply(count_non_zero_or_false)

row_counts_df = pd.DataFrame({'Feature': features_filtered.columns, 'Non-
Zero/False Count': row_counts})

features_to_keep = row_counts_df[row_counts_df['Non-Zero/False Count'] >
5]['Feature']

exclude_keywords = ['igg', 'iga', 'igm']
final_features = [
```

[59] See Section 7.2.1 *Hospital Admission Risk Factors Identification*.

```
    feature for feature in features_to_keep
    if not any(keyword in feature.lower() for keyword in exclude_keywords)]

features_final = features_filtered[final_features]

X_train, X_test, y_train, y_test = train_test_split(features_final, target,
test_size=0.3, random_state=14)
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)

y_pred = log_reg.predict(X_test)

print("Classification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("Accuracy Score:")
print(accuracy_score(y_test, y_pred))
```

For the features, calculate and present coefficients and odds ratio.

```
final_feature_names = features_final.columns

coefficients = log_reg.coef_[0]

coef_df = pd.DataFrame({
    'Feature': final_feature_names,
    'Coefficient': coefficients})

coef_df['Absolute Coefficient'] = coef_df['Coefficient'].abs()
coef_df['Odds Ratio'] = np.exp(coef_df['Coefficient'])

def count_non_zero_or_false(column):
    if column.dtype == 'object':
        return (column != 'False').sum()
    else:
        return (column != 0).sum()

row_counts = features_final.apply(count_non_zero_or_false)

row_counts_df = pd.DataFrame({'Feature': features_final.columns, 'Non-
Zero/False Count': row_counts})

coef_df = coef_df.merge(row_counts_df, on='Feature', how='left')
coef_df = coef_df.sort_values(by='Odds Ratio', ascending=False)
coef_df
```

Appendix I – Logistic Regression for Enhanced Prediction

This appendix outlines the Python code used to build a logistic regression model, enhancing the predictive accuracy of the previous model to better forecast hospital admissions.

Carry out hyperparameter tuning and select the best parameters for the model that is built and evaluated.[60]

```
features_final = features_filtered[final_features]

X_train, X_test, y_train, y_test = train_test_split(features_final, target,
test_size=0.3, random_state=14)

f1_scorer = make_scorer(f1_score, pos_label=1)

param_grid = {'C': [0.01, 0.1, 1, 10, 100, 1000], 'penalty': ['l1', 'l2',
'elasticnet'], 'solver': ['liblinear', 'saga']}
grid_search = GridSearchCV(LogisticRegression(max_iter=1000), param_grid,
cv=5, scoring=f1_scorer)
grid_search.fit(X_train, y_train)
best_model = grid_search.best_estimator_

print("Best Parameters:", grid_search.best_params_)
print("Best F1 Score:", grid_search.best_score_)

y_pred = best_model.predict(X_test)

print("Classification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("Accuracy Score:")
print(accuracy_score(y_test, y_pred))

f1 = f1_score(y_test, y_pred, pos_label=1)
precision = precision_score(y_test, y_pred, pos_label=1)
recall = recall_score(y_test, y_pred, pos_label=1)

print(f"F1 Score: {f1:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

[60] See Section 7.2.2 *Hospitalisation Prediction*.