Exploring Patterns in Nuclear Physics Data Through Machine Learning

Karen Weider

MSc by research

University of York

School of Physics, Engineering and Technology

April 2024

1 Abstract

This thesis explores the application of machine learning algorithms to nuclear physics data, aiming to uncover patterns within the data and reveal relationships between various nuclear characteristics. While the research demonstrated some success, particularly in identifying correlations between separation energies, shell models, and magic numbers, it also encountered significant challenges. The most significant among these was the limitation posed by the quality and quantity of available data, which affected the accuracy and reliability of predictions, such as those for proton and neutron drip lines.

The research adopted a broad, exploratory approach, intentionally avoiding the use of established physics models to allow machine learning to independently identify patterns. However, this wide-ranging focus, combined with data limitations, resulted in findings that are insightful but often inconclusive. The experiments conducted, including attempts to relate nuclear deformity to stability and to apply machine learning to a model influenced by the polyspheron model, further underscored the need for better and more targeted data.

This thesis highlights the potential of machine learning in nuclear physics but also emphasises the importance of depth and data quality in future research. The results provide a foundation for more focused studies, where improved datasets and a narrower research scope could yield more definitive insights.

Contents

List of Tables

List of Figures

2 Author’s Declaration

I, Karen Weider, declare that this thesis is a presentation of original work, and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

3 Research Objective

This thesis embarks on an exploratory investigation into the application of machine learning techniques to nuclear physics data. The primary objective is to discover patterns and relationships within the data without relying on existing theoretical models, allowing the machine learning algorithms to identify correlations independently. By focusing on a data-driven approach, this research aims to provide fresh insights into nuclear phenomena, exploring areas such as the prediction of proton and neutron drip lines, the relationships between separation energies, shell models, and magic numbers, as well as the connections between nuclear deformity and stability.

Throughout this investigation, the role of data quality and scope in influencing predictive accuracy is critically examined. The thesis also includes the application of machine learning to a novel nuclear core model, inspired by the polyspheron model, to assess the algorithm's ability to make predictions in areas where traditional physics models are not available. This theoretical framework sets the stage for a detailed analysis of machine learning’s potential to contribute to nuclear physics, highlighting both the challenges and opportunities presented by this interdisciplinary approach.

4 Chapter 1: Background

4.1 Nuclear Physics

4.1.1 Introduction to Nuclear Physics

Nuclear physics investigates the fundamental properties and behaviours of atomic nuclei, focusing on critical aspects such as stability, decay processes, and interactions. The insights gained from this field are pivotal not only for understanding the origins and evolution of the universe but also for practical applications across various industries, including medicine, defence, and energy production. Traditionally, nuclear physics research has been grounded in theoretical models like the liquid drop model and the shell model, which have been refined through decades of experimental validation. However, the inherent complexity of nuclear systems and the vast number of potential interactions present significant challenges for conventional analytical methods.

4.1.2 Magic Numbers

Magic numbers refer to certain values of protons or neutrons (or both) that result in enhanced stability within atomic nuclei. These magic numbers correspond to filled nuclear shells, similar to electron shells in atoms. They play a crucial role in determining the stability and properties of atomic nuclei. They are 2, 8, 20, 28, 50, 82 and 126.

4.1.3 Major Shell Closures

A major shell closure refers to a configuration in which a nucleus has a complete shell of either protons or neutrons, corresponding to specific numbers known as "magic numbers." These numbers are 2, 8, 20, 28, 50, 82, and 126. When a nucleus reaches a major shell closure, it exhibits exceptional stability due to the complete filling of energy levels within the nuclear shell model. At major shell closures, nuclei tend to be spherical. This spherical symmetry arises because the filled shells minimise the energy of the system, reducing any deformation.

4.1.4 Minor Shell Closures

Minor shell closures, sometimes referred to as semi-magic numbers, represent points within the nuclear shell model where nucleons (protons or neutrons) form partial or subshells that confer a degree of enhanced stability, albeit less pronounced than that found at major shell closures (the magic numbers). Typical minor shell (up to N = 40) closure numbers for neutrons include 6, 14, 16, 32, 38, and 40.

4.1.5 Bosons

Bosons are particles with integer values of spin (0, 1, 2, etc.). They obey Bose-Einstein statistics and can occupy the same quantum state simultaneously. Examples of bosons include photons (the particles of light), W and Z bosons (mediators of the weak nuclear force), and the Higgs boson.

4.1.6 Fermions

Fermions, on the other hand, have half-integer values of spin (1/2, 3/2, etc.). They obey Fermi-Dirac statistics and follow the Pauli Exclusion Principle, meaning that no two fermions can occupy the same quantum state simultaneously. Fermions include particles like electrons (constituents of atoms), quarks (which make up protons and neutrons), and neutrinos.

4.1.7 Separation Energy (Sn or Sp)

Separation energy refers to the energy required to remove a particle or a group of particles from a nucleus. The separation energy can apply to neutrons, protons, alpha particles, and even larger fragments, depending on the context. The primary types of separation energies are neutron separation energy (Sn) and proton separation energy (Sp). The greater the separation energy, the more energy is required to remove a particle, indicating a more stable nucleus. Conversely, a low separation energy suggests a less stable nucleus, more prone to radioactive decay.

4.1.8 Shell Model

The shell model is a fundamental concept in nuclear physics that describes the structure and behaviour of nucleons (protons and neutrons) within an atomic nucleus. In the shell model, nucleons are said to occupy a series of discrete energy levels or "shells" within the nucleus, similar to electrons in atomic orbitals. These shells are filled according to the Pauli exclusion principle, which states that no two nucleons can occupy the same quantum state simultaneously.

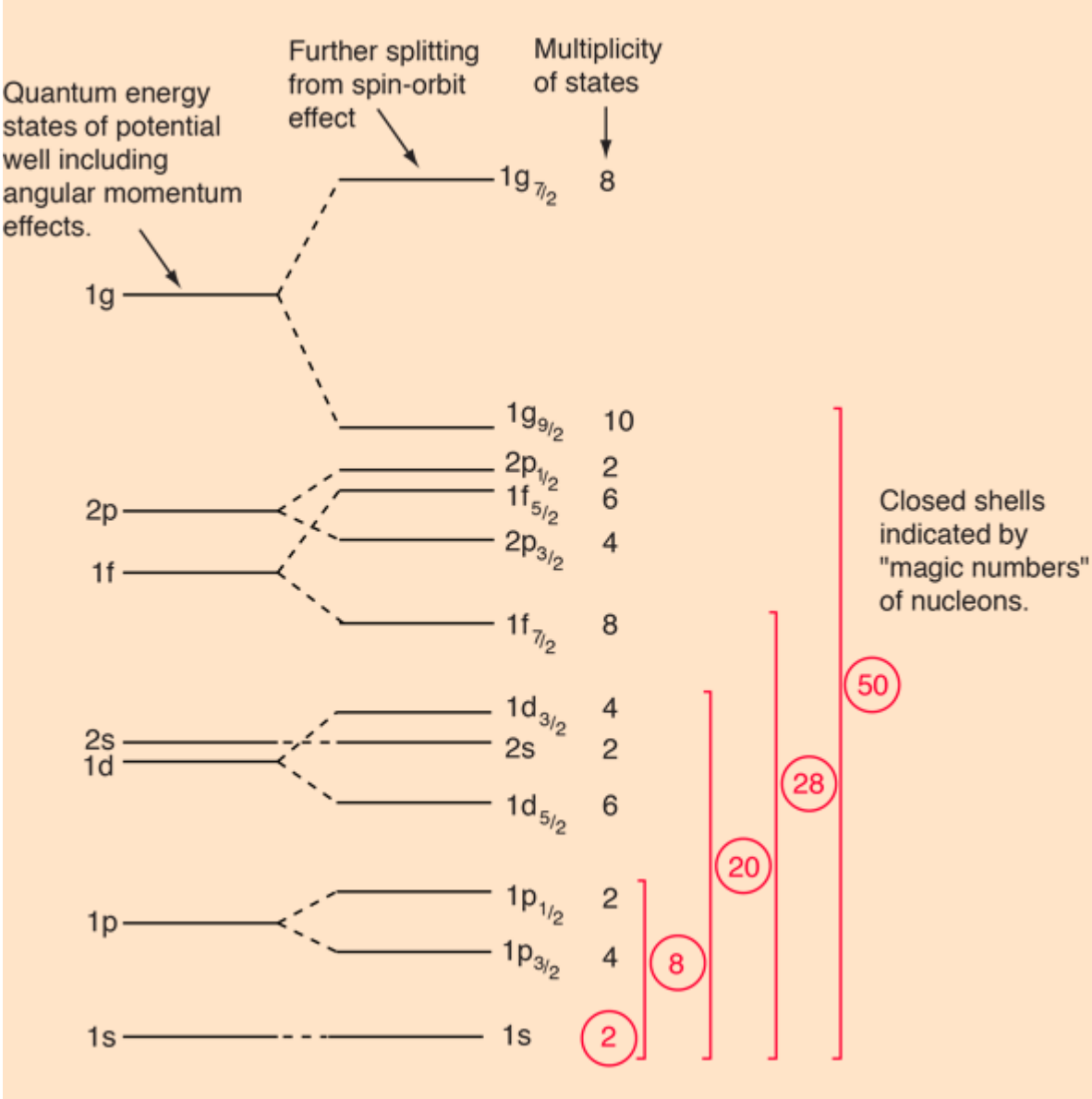


Figure 1: Shell Model

4.1.9 Spin

Spin is a fundamental property of elementary particles in quantum mechanics, representing their intrinsic angular momentum.

4.1.10 Deformity

Atomic nuclei are not perfectly spherical. They can exhibit various shapes, including spherical, prolate (elongated), oblate (flattened), or even triaxial (irregular). These different shapes arise from the interplay of nuclear forces and the quantum-mechanical behaviour of nucleons (protons and neutrons) within the nucleus.

4.1.11 Energy Levels

Energy levels represent the possible values of energy that a nuclei can have. They are related to the arrangement of protons and neutrons within them. These energy levels contribute to phenomena such as nuclear stability, radioactive decay, and nuclear reactions.

#	Nuclide	E_x [keV]	J^{π} order	Band	$T_{1/2}$	$T_{1/2}$ [s]	Decay modes BR [%]	Isospin	μ [μ_N]	Q [b]
1	$^{18}_{7}N_{11}$	0.0	1-		619 ms *2*	0.619 *2*	β- 100 β- α 12.2 *6* β- n 12.0 *13*	2	0.3273 *4*	0.0123 *12*
2	$^{18}_{7}N_{11}$	114.71 *10*	(2-)		0.40 ns *11*	400E-12 *11*	IT 100			
3	$^{18}_{7}N_{11}$	587.39 *20*	(2-) *2*				IT 100			
4	$^{18}_{7}N_{11}$	742.0 *4*	(3-)				IT 100			
5	$^{18}_{7}N_{11}$	1170 *20*	(1-)				IT 100			

Figure 2: Energy Levels

4.1.12 Energy Levels and Magic Numbers

The energy levels are determined by the nuclear potential, which is typically modelled as a combination of a strong, attractive nuclear force and a centrifugal term due to the nucleon's angular momentum. As protons or neutrons fill these shells, certain numbers of nucleons (known as "magic numbers") correspond to especially stable configurations. These magic numbers are 2, 8, 20, 28, 50, 82, and 126 and are observed experimentally as points at which nuclei exhibit extra stability.

4.1.13 Neutron Drip Line

The "neutron drip line" is a concept in nuclear physics that identifies the boundary at which a nucleus can no longer hold additional neutrons. Nuclei beyond this line are unstable against neutron emission, meaning that if you try to add more neutrons, they will not be bound within the nucleus and will simply "drip" out. This line essentially defines the limit of how neutron-rich a stable isotope can be.

Nuclei near or beyond the neutron drip line are extremely neutron-heavy compared to their number of protons. These isotopes are inherently unstable and tend to undergo decay quickly, often through processes such as beta decay where a neutron is transformed into a proton, an electron, and an anti-neutrino.

The neutron drip line is not a fixed line and varies significantly across different elements. It is much less well-defined than the proton drip line because the neutrons do not repel each other through electromagnetic force (as protons do), allowing a greater accumulation of neutrons

before reaching instability. This makes the location of the neutron drip line more difficult to determine and less predictable than the proton drip line.

Understanding where the neutron drip line lies for various elements helps in exploring the limits of the nuclear landscape and has implications in astrophysics, particularly in the study of neutron-rich environments like those found in neutron stars or during certain types of stellar explosions and nucleosynthesis processes.

The neutron drip line can be seen in figure 3.

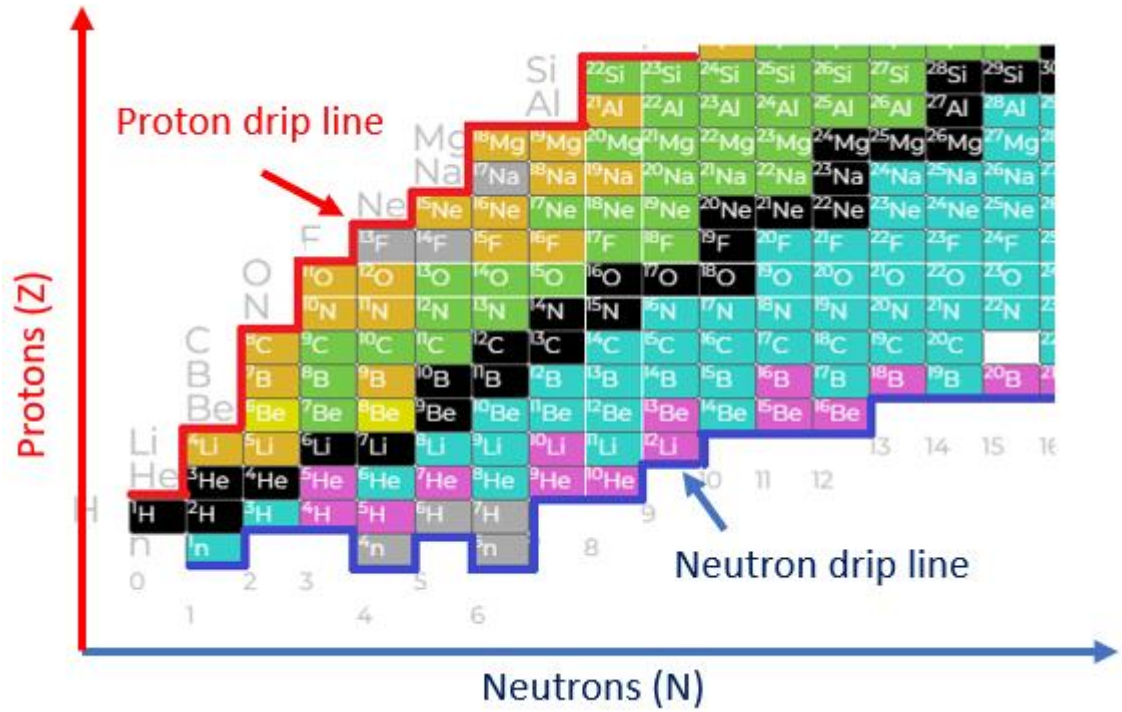


Figure 3: Proton & Neutron Drip Line

4.1.14 Proton Drip Line

The "proton drip line" is a concept in nuclear physics referring to the boundary beyond which nuclei are no longer stable against the emission of protons. Nuclei situated on this boundary or beyond it are so proton-rich that they can spontaneously emit protons.

In simpler terms, if you add more protons to a nucleus that is at the proton drip line, the resulting nucleus will not hold together; the additional protons will "drip" out, as the nuclear force that binds the nucleus is insufficient to counterbalance the electromagnetic repulsion between the excessive number of protons.

This concept is analogous to the "neutron drip line," which defines the boundary for neutron-rich nuclei beyond which additional neutrons are not bound and will also "drip" out.

These drip lines help scientists understand the limits of nuclear stability and the range of possible isotopes. The study of nuclei near the drip lines is significant in both theoretical

nuclear physics and astrophysics, as it contributes to our understanding of nucleosynthesis in stars and cosmic phenomena. The proton drip line can be seen in figure 3.

4.2 Artificial Intelligence and Machine Learning

4.2.1 Introduction to Artificial Intelligence.

Artificial intelligence (AI) represents a broad field of computer science focused on creating systems capable of performing tasks that typically require human intelligence. These tasks include problem-solving, pattern recognition, and decision-making. Within AI, various subfields focus on specific capabilities, such as natural language processing, computer vision, and robotics. AI has increasingly been integrated into scientific research, offering powerful tools for analysing large datasets and uncovering insights that might be missed by traditional methods.

4.2.2 Introduction to Machine Learning

Machine learning (ML) is a subset of Artificial Intelligence (AI) that specifically deals with the development of algorithms capable of learning from and making predictions based on data. Unlike traditional programming, where explicit instructions are given, ML algorithms improve their performance as they process more data, identifying complex patterns and relationships. ML can be applied in either a supervised manner, where the algorithm is trained on labelled data, or in an unsupervised manner, where the algorithm independently identifies patterns in unlabelled data.

The emergence of machine learning provides new avenues for exploring complex scientific data, offering the potential to reveal hidden patterns, identify novel correlations, and enhance our understanding of intricate phenomena such as those found in nuclear physics.

Historically, nuclear research relied on theoretical models refined through experimentation, a laborious process hindered by manual data analysis. Although advancements in experimental technology allowed for the generation of large datasets, computing limitations hindered meaningful analysis. Machine learning facilitates quicker analysis, explores theories more efficiently, and uncovers data trends and patterns without preconceived notions.

For more information on the machine learning process please see chapter 5: Methodology.

5 Chapter 2: Methodology

The machine learning process involves a series of systematic steps that transform raw data into predictive models. These steps guide the development and training of models to accurately forecast outcomes based on input data. Understanding this process is important for implementing effective machine learning solutions.

5.1 Define the Computing Environment

5.1.1 Python

Python is a versatile programming language known for its simplicity and readability. It offers a wide range of libraries and frameworks, making it suitable for various tasks, including data analysis, machine learning, and scientific computing.

5.1.2 Anaconda Environment

Anaconda is a distribution of Python that comes bundled with many pre-installed libraries and tools commonly used in data science and scientific computing. It provides a convenient way to manage Python environments and packages, ensuring compatibility and reproducibility across different projects.

5.1.3 Python Packages

Below is an overview of the most common packages used. Due to the trial-and-error nature of the machine learning process the exact packages could not be defined at the start of an experiment. The exact set of packages for each experiment were decided as the experiment evolved.

- Pandas: Pandas is a powerful library for data manipulation and analysis in Python. It provides data structures like DataFrame and Series, which are ideal for handling structured data, such as tables and time series.
- NumPy: NumPy is a fundamental package for numerical computing in Python. It provides support for multi-dimensional arrays, mathematical functions, linear algebra operations, and random number generation, making it essential for scientific computing tasks.
- Matplotlib: Matplotlib is a plotting library for creating static, interactive, and animated visualizations in Python. It offers a wide range of plotting functions

and customization options, making it suitable for creating publication-quality figures for data analysis and presentation.

- Scikit-learn: Scikit-learn is a popular machine learning library in Python, providing simple and efficient tools for data mining and analysis. It includes various algorithms for classification, regression, clustering, dimensionality reduction, and model selection, along with utilities for preprocessing and model evaluation.
- Pickle: Pickle is a module in Python used for serializing and deserializing Python objects. It allows you to save the state of your Python objects to disk and reload them later, making it useful for saving machine learning models and other complex data structures.
- Matplotlib.pyplot: Pyplot is a sub-module of Matplotlib that provides a MATLAB-like interface for creating plots and visualizations in Python. It is commonly used for quick and easy plotting tasks, such as creating scatter plots, histograms, and line plots.
- Seaborn: Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics, making it easy to visualise complex relationships in your data.
- Ipywidgets: IPywidgets is a library for creating interactive widgets and controls in Jupyter notebooks. It allows you to add interactive elements like sliders, buttons, and dropdown menus to your notebooks, enabling users to interactively explore and analyze data.
- Keras: Keras is a high-level neural networks API written in Python, capable of running on top of TensorFlow, CNTK, or Theano. It enables fast experimentation with deep neural networks and provides easy-to-use interfaces for building and training neural networks.
- RandomForest: RandomForest is an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes or mean prediction of the individual trees.

5.2 Define the Experiment Objective

Each machine learning (ML) experiment begins with a clearly defined objective that originates from a specific research question. For example, if the question is, “What nuclear properties influence spin?”, the objective would be to identify the relationships between these nuclear properties and spin. A well-defined objective is crucial as it guides the selection of the appropriate ML model. The nature of the objective determines the type of task to be performed, which in turn influences the choice of the model and approach needed to achieve the desired outcome.

To effectively select a machine learning model, the experiment's objective must align with the type of task required. The task may involve classifying data into categories, predicting numerical values, detecting anomalies, or grouping similar data points. The nature of the task helps in identifying the most suitable algorithms.

Consider the following types of tasks:

- **Classification Tasks**: These tasks involve assigning data points to predefined categories. Models commonly used for classification include decision trees, support vector machines, and neural networks.
- **Regression Tasks**: Regression tasks focus on predicting continuous numerical values. Suitable models for regression include linear regression, random forests, and gradient boosting machines.
- **Clustering Tasks**: Clustering involves grouping similar data points without predefined labels. Models such as k-means and hierarchical clustering are often employed for this purpose.

Aligning the experiment’s objective with the appropriate task type ensures a more effective selection of the machine learning model, thereby enhancing its capability to address the research question effectively.

5.3 Identify Data Sources and Clean Data

The same data was reused for the majority of the experiments.
The sources were as follows:

1. Chart of Nuclides (ChartofNuclides)

   The following datasets were downloaded from the Nuclear Data Section (NDS) of the International Atomic Energy Agency (IAEA). Extensive data cleaning operations were performed to address inaccuracies and inconsistencies.

   - Separation energies
   - Excitation energies
2. NuDat (NuDat)
   - Deformation data

Before any machine learning models can be developed, the data must be cleaned. Data cleaning involves preparing and refining the dataset to ensure accuracy, consistency, and relevancy before it is used for processing and analysis. This involves actions to handle data issues such as,

- Unreadable characters.
- Blank spaces.
- Missing values.
- Converting data types.
- Replacing labels.
- Normalizing data.
- Uneven distributions.

Each time new data is introduced it needs to be analysed and cleaned. Effective data analysis ensures that the dataset is clean and well-prepared, setting a strong foundation for building a reliable machine learning model.

The datasets initially contained a significant amount of unusable data, requiring extensive cleaning operations before it could be used in a machine learning algorithm. This cleaning process significantly reduced the size of the datasets, which in turn impacted the performance of the machine learning algorithm and, consequently, the reliability of the results.

5.3.1 Addressing skewed data

The shell closure values calculated in experiment 3 in this thesis appeared skewed because the likelihood of a value appearing is not uniform across all possible values. Some values, such as 82, are more likely to appear as a shell closure than others, like 2, because more datasets contain instances of 82.

To address this imbalance, a normalization function was developed. This function adjusts the frequency of each shell closure value by dividing the number of times a specific value appears as a shell closure by the total number of occurrences of that value in the dataset.

For example, if N = 2, there is limited data containing this value. According to the Chart of Nuclides, only 6 nuclei have N = 2, meaning there are 6 possible opportunities for it to be identified as a shell closure. If N = 2 appears 6 times in the results, the normalization (or confidence level) would be 100%.

On the other hand, for N = 82, there are 29 possible occurrences in the Chart of Nuclides. If it appears 6 times in the results, the confidence level would be 0.2 (or 20%).

By using these confidence levels rather than just raw frequencies, we can achieve a more balanced and accurate analysis of the data, focusing on values with a sufficient degree of confidence.

5.3.2 Addressing Class Imbalance

Class imbalance happens when the data groups are unevenly populated for example a data set may contain two classes such as stable and unstable nuclei. This was seen in experiment 4. Naturally, there are more unstable than stable nuclei, so the data set is very imbalanced, in favour of the unstable nuclei. This causes several issues.

- Majority Class Bias: Machine learning algorithms tend to be biased towards the majority class, meaning they prioritise accuracy on the majority class at the expense of the minority class. As a result, the model may have a tendency to classify instances into the majority class, leading to poor performance on the minority class.
- Misleading Evaluation Metrics: Traditional evaluation metrics like accuracy may not be reliable in the presence of class imbalance. For instance, a model that predicts all instances as the majority class can achieve high accuracy if the majority class dominates the dataset. Therefore, accuracy alone is not a good indicator of model performance.
- Impact on Model Learning: Class imbalance can affect the learning process of machine learning algorithms. Models may struggle to learn the minority class patterns effectively

due to their limited representation in the dataset. Consequently, the model may fail to generalise well to unseen data, especially for the minority class.

To address class imbalance, various techniques can be employed, including:

- Resampling: Oversampling the minority class or under sampling the majority class to balance the dataset.
- Algorithmic Techniques: Using algorithms specifically designed to handle class imbalance, such as cost-sensitive learning or ensemble methods like SMOTE (Synthetic Minority Over-sampling Technique).
- Evaluation Metrics: Focusing on evaluation metrics that are more informative in the presence of class imbalance, such as precision, recall and F1-score.
- Stratification is used in machine learning to ensure that the distribution of classes in the training and testing datasets remains similar. This is particularly important when dealing with imbalanced datasets, where one class may be significantly more prevalent than others. This was used in experiment 5.

Understanding the domain context is important when dealing with class imbalance. In some cases, the imbalance may reflect the natural distribution of classes in the real-world scenario as for the case of nuclear stability. Therefore, it's essential to consider domain knowledge when deciding on the appropriate approach to handle class imbalance.

5.4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an activity carried out on the cleaned data which includes exploring the dataset to understand its characteristics, identify patterns, detect anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations. It is always a good idea to do some EDA before launching into machine learning because a lot can be revealed about data once it can be visualised.

5.4.1 Data Set Types

Data sets can be labelled or unlabelled. An unlabelled data set has no labels i.e. no data descriptions. It is used in unsupervised learning algorithms. Table 1 is an example of unlabelled data.

4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5	3.4	1.5	0.2

Table 1: Unlabelled Data

Unsupervised learning is a type of machine learning that deals with unlabelled data. The primary aim of unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data itself, without any guidance from a known output.

A labelled dataset is a dataset where each piece of data is paired with a specific label or annotation that describes or categorises it. These labels serve as the "answers" or "outputs" for each data entry, which a machine learning model uses to learn from during training. The purpose of having labels is to guide the learning algorithm in understanding the relationships between the input features and the desired output, enabling it to predict or categorise new, unseen instances based on the learned patterns. The data below is labelled, it has a column called ‘species’ which is how you would categorise data with similar sepal_length, sepal_width, petal_length and petal_width values.

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Table 2: Labelled Data

Supervised learning is a type of machine learning that deals with labelled dataset and is used to predict output values based on a set of input values.

Understanding whether the ML model will be supervised or unsupervised is necessary before deciding which type of model to use.

5.4.2 Feature Selection and Feature Engineering

A dataset will contain one or more features. A feature is a measurable quantity in the data. For example, in table 2, sepal_width is a feature. Before building an ML model the features must be chosen. Which features and how many features can influence the outcome, so this step is often revisited during the evaluation step to improve results. After an initial data analysis or after trying out an initial ML model, it may be necessary to add further features. This is called feature engineering. Feature engineering is the process of adding new columns of data (features) to an existing data set. The new data is usually derived from the existing data from adding, subtracting or applying a function to existing data. It usually happens in the data pre-processing stage.

5.5 Problem Definition and Objective Setting

Before selecting a machine learning model, it's important to first identify the type of problem that needs to be solved. In machine learning, problems are generally categorised based on the nature of the task. The key types of machine learning problems addressed in this thesis are:

- Regression Problems: These involve predicting a continuous numeric value. The goal is to estimate a variable that can take any real number, such as predicting house prices, temperature changes, or stock market values.
- Classification Problems: Classification involves predicting discrete categories or classes. The model learns to assign data points to predefined labels. Examples include determining whether an email is spam or not or classifying images into categories like cats and dogs.
- Clustering Problems: Clustering is a form of unsupervised learning where the objective is to group similar data points into clusters without predefined labels. The algorithm discovers inherent structures in the data, such as segmenting customers based on purchasing behaviour or grouping similar documents together.

Each type of problem requires different techniques and models to provide effective solutions.

5.6 Initial Model Selection

Once the data has been cleaned, prepared and examined, and the problem and objective have been clearly defined, the next step is selecting the appropriate ML model. The choice of model depends on the nature of the experiment and the specific goals to be achieved. (e.g., classification, regression) and the data characteristics. Factors such as the accuracy,

interpretability, complexity, and the computational efficiency of the model are considered. Sometimes, multiple models are tested in parallel during this step to identify the most effective approach. The initial model is only a starting point and after much experimenting and tuning, a different model may be selected.

5.6.1 Classification Model Selection

Classification models in machine learning are models that learn to predict the class labels of input data points. They are used when the output variable is a category, such as "spam" or "not spam" for email classification, or "cat," "dog," or "bird" for image classification. These models analyse the features of the data and learn patterns to classify new instances into predefined categories.

Below is a simplified overview of some common classification models.

- Logistic Regression: Despite its name, logistic regression is a classification model used for binary classification. It models the probability that a given input belongs to a certain class.
- Decision Trees: Decision trees split the data into subsets based on features and create a tree-like structure of decisions. Each node represents a feature, each branch a decision based on that feature, and each leaf node a class label.
- Random Forests: Random forests are an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.
- Support Vector Machines (SVM): SVMs find the hyperplane that best separates classes in feature space. They work well in high-dimensional spaces and are effective for both linear and non-linear classification tasks.
- Naive Bayes: Naive Bayes classifiers are based on Bayes' theorem with an assumption of independence between features. Despite their simplicity, they are effective for many classification tasks, especially in text classification.
- K-Nearest Neighbours (KNN): KNN classifies new data points based on the majority class among their k nearest neighbours in the feature space.

Neural Networks: Neural networks, particularly deep learning models, consist of layers of interconnected nodes that learn complex patterns from the data. They have achieved state-of-the-art performance in various classification tasks.

5.6.2 Clustering Model Selection

Clustering is a type of unsupervised machine learning technique used to group data in such a way that data in the same group (called a cluster) are more like each other than to those in other groups. It needs no prior knowledge of the group labels. The choice of clustering algorithm is influenced by the specific characteristics of the data, the objective of the experiment, and the desired outcome of the clustering process.

Common clustering algorithms include,

- K-Means: This algorithm partitions the data into K distinct, non-overlapping clusters. It assigns each data point to the closest cluster by minimizing the sum of the squared distances between the data points and their respective cluster's centroid. It's ideal when you have a good estimate of the number of clusters and expect them to be roughly equal in terms of the number of data points. K-Means is widely used due to its simplicity and computational efficiency.
- Hierarchical Clustering: This method builds a hierarchy of clusters either through a bottom-up approach (agglomerative) or a top-down approach (divisive). It is beneficial for data where the relationships between clusters are hierarchical or nested.
- Mean Shift: This algorithm locates the centres of clusters without assuming the number of clusters beforehand. Starting from each data point, Mean Shift iteratively moves towards the region with the highest density of points (the mode of the point density function) until convergence. It finds clusters of varying shapes and sizes, making it versatile for complex data.
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): This algorithm identifies clusters as areas of high density separated by areas of low density. Points in low-density regions are classified as noise. DBSCAN does not require specifying the number of clusters; it infers them from the data based on the density. The key parameters are eps, which determines the search radius for nearby neighbours, and min_samples, which defines how dense a region should be to form a cluster.
- Agglomerative Clustering: This is a hierarchical clustering method that builds clusters by merging pairs of data points or existing clusters based on their proximity. The process begins by treating each data point as a single cluster and then successively merges pairs of clusters until all points belong to a single cluster or a stopping criterion is met. Agglomerative clustering can reveal complex structures and is particularly useful for understanding the hierarchical relationships in data.

The selection of a clustering model also involves trial and error. This includes tuning model parameters, such as the number of clusters for K-Means or the density parameters for DBSCAN, and evaluating the results using metrics such as silhouette score or within-cluster sum of squares. By comparing the performance of different algorithms and assessing how well they meet the objectives of the experiment, the model that best captures the underlying structure of the data can be identified.
Through this iterative process, the optimal clustering model can be selected to effectively reveal patterns and relationships within the data.

5.6.3 Neural Networks

For highly complex problems that require pattern recognition from large datasets, neural networks or deep learning models are often more suitable than simpler models. These techniques are chosen when traditional methods are insufficient for capturing complex relationships in the data.
A neural network is a series of algorithms designed to recognise underlying patterns in a dataset by mimicking the way the human brain operates. It is composed of several layers:

- Input Layer: This is where the data enters the system. It functions like the senses of the neural network, receiving and processing the raw input data.
- Hidden Layers: These layers are where most of the computation occurs. Neurons in the hidden layers process the data by learning patterns and relationships, like the thought process in the human brain.
- Output Layer: This layer provides the final output, acting as the decision-making or predictive element of the neural network.

A Deep Neural Network (DNN) is a specialised type of neural network that includes multiple hidden layers between the input and output layers. These additional layers enable the network to learn and capture more complex patterns and relationships within the data, making DNNs particularly useful for tasks such as image recognition, speech processing, and natural language understanding.

5.6.4 Regression Model Selection

During the process of selecting the appropriate regression model for experiment 1, the initial selection was a simple linear regression model, e.g. scikit-learn LinearRegression, this was used to establish a baseline. This approach provided a clear starting point and enabled the evaluation of the fundamental relationship between input features and the target variable.

When it became apparent that the data exhibited non-linear relationships or complex interactions that a linear model could not adequately capture, the focus shifted to more sophisticated models. Decision trees and random forests were explored for their ability to model intricate patterns and interactions without extensive feature engineering. For datasets characterised by high-dimensional features or complex non-linear relationships, neural networks were also considered. These models demonstrated the capacity to learn complex functions and interactions, though they required substantial data and meticulous tuning to mitigate overfitting.

By initially employing simpler models and progressively advancing to more complex ones as necessary, a systematic approach was taken to identify the model that best suited the data and fulfilled the experiment's objectives.

5.6.5 Model Summary

Table 3 shows a summary of the experiments and model selected.

Experiment	Problem type	Data type	Selected model
1: Predicting masses	Regression	Unlabelled	Unsupervised SVM
2: Predicting separation energy	Regression	Unlabelled	Unsupervised neural network
3: Predicting drip line	Clustering	Unlabelled	Unsupervised K-Means
4: Predicting stability using energy density	Binary classification	Labelled	Supervised Random Forest
5: Predicting stability using energy densities	Binary classification	Labelled	Supervised Random Forest
6: Predicting spin	Data analysis	NA	NA
8 predicting deformity	Regression	Unlabelled	Unsupervised random forest regressor

Table 3: Summary of experiments and selected models

5.7 Model Training

The data is split into two sets called training and test. The percentage of data each set is another variable and can affect the quality of the output results. The learning occurs on subset called the training data. During this stage, the model attempts to find patterns or relationships within the data that correlate inputs to their respective outputs. The model adjusts its internal parameters or weights through an optimisation process. The process is unique to the model selected. This adjustment is essential as it allows the model to learn from the data—hence, this is referred to as the "learning" phase.
The effectiveness of this process hinges on the use of algorithms that can iteratively improve the model's predictions by minimising errors, typically measured by a loss function.

5.8 Model Test and Validation

After the model has been trained, it is then tested on a separate subset of the data known as the test data. This data is unseen by the model during the training phase, which ensures that the testing process evaluates how well the model can generalise its learned patterns to new, unknown datasets.

5.9 Model Evaluation

Model evaluation involves comparing the model's predictions on the test data against the actual outcomes. This comparison helps to assess the accuracy of the model, indicating how well it has learned and predicted the underlying relationships within the data. Accurate models show minimal discrepancy between predicted and actual results, signifying successful learning and effective generalisation. The model valuation techniques vary according to the type of model used.

5.9.1 Z-Score

A Z-score is a statistical measurement that describes a value's relationship to the mean of a group of values. It is measured in terms of standard deviations from the mean. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. A Z-score of 1.0 would indicate a value that is one standard deviation from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean.

5.9.2 Regression Model Evaluation

5.9.2.1 R^2 value

The R^2 value provides an indication of goodness of fit and tells you how well the data fits the statistical model. It is only applicable in linear regression.

5.9.2.2 Mean Squared Error (MSE) and Root Mean Squared Error (RMSE)

For regression problems, the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are popular evaluation metrics. MSE calculates the average squared difference between predicted and actual values, penalizing larger errors more heavily. RMSE is the square root of MSE, offering an interpretable error value in the same unit as the target variable. These metrics provide a direct measure of how far off the predictions are from actual outcomes.

5.9.2.3 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) calculates the average of the absolute differences between predicted and actual values. Unlike MSE, which emphasises larger errors, MAE gives equal weight to all errors. It is a useful metric for understanding the overall magnitude of errors in a model's predictions.

5.9.3 Clustering Model Evaluation

The following metrics can be used to evaluate the performance of a clustering model.

- Silhouette Score: Measures how similar an object is to its own cluster compared to other clusters. The silhouette values range from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

- Davies-Bouldin Index: The ratio of within-cluster scatter to between-cluster separation. The lower the Davies-Bouldin index, the better the clustering is considered, as it implies a high inter-cluster distance and low intra-cluster distance.

- Calinski-Harabasz Index: Also known as the Variance Ratio Criterion, this index is the ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters. The higher the score, the better defined the clusters are.

Visual assessments should also be made where possible as they can provide intuitive insights into how well the clustering process has performed.

Below are the results taken from a comparison between several different clustering models. They were compared using the metrics above and by visually comparing the quality of the clustering.

	Silhouette Score	Davies-Bouldin Score	Calinski-Harabasz Score
KMeans	0.619712	0.408437	62.679430
Agglomerative	0.619712	0.408437	62.679430
DBSCAN	0.396997	7.381505	12.876503
MeanShift	0.486718	0.481069	62.508792

Figure 4: Comparison scores to aid model selection

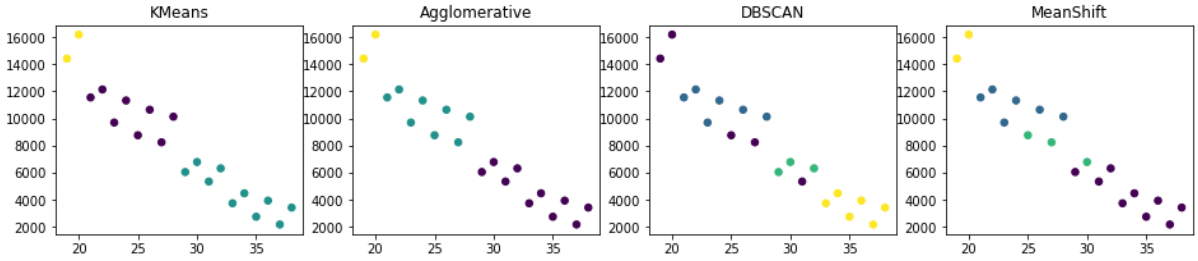


Figure 5: Visual cluster comparison to aid model selection

Figure 5 shows the data taken from the clustering experiment 3 and shows how you can use visual comparisons of each model result to identify the best model to use.
As can be seen in the figures 4 and 5, K-Means and Agglomerative achieved the highest scores. K-Means was selected due it's speed and ease of implementation.

5.9.4 Classification Model Evaluation

The following metrics can be used to evaluate model performance,

- Accuracy: Accuracy measures the ratio of correctly predicted instances to the total number of instances in the dataset. It's the most intuitive metric and indicates overall model performance. However, it may not be suitable for imbalanced datasets.
- Precision: Precision measures the ratio of true positive predictions to the total number of positive predictions made by the model. It indicates the model's ability to correctly identify positive instances without falsely classifying negative instances as positive.

- Recall (Sensitivity): Recall measures the ratio of true positive predictions to the total number of actual positive instances in the dataset. It indicates the model's ability to identify all positive instances correctly.
- F1-Score: F1-Score is the harmonic mean of precision and recall. It provides a balance between precision and recall, especially when dealing with imbalanced datasets. F1-Score is useful when you want to seek a balance between precision and recall.
- Confusion Matrix: A confusion matrix provides a detailed breakdown of a model's classification performance, showing true positives, true negatives, false positives, and false negatives. This can be seen in experiment 6.

Model	Accuracy	Precision	Recall	F1-Score	Correctly predicted stable	Correctly predicted stable
Random Forest	0.93	0.95	0.97	0.96	22 (34.92%)	756 (97.30%)
KNN	0.89	0.93	0.96	0.94	8 (12.70%)	743 (95.62%)

Table 4: Table comparing performance between models.

It was harder to visually assess these models because I wanted to focus on stable and the data was overwhelmingly unstable. I compared their metrics and how many stable were correctly predicted and decided the random forest was the best model in this situation.

5.10 Model Improvement

To improve the evaluation data and overall results, the model evolves through a series of improvements. This is the most timing consuming part of machine learning. Deciding on what improvements to make can be done as trial and error and in some situations, tests scripts can be created to modify values through a specific range revealing what values return the best results.

5.10.1 Changing Python Packages

In some cases, improving the precision of the arithmetic can help. High precision arithmetic
In this thesis, mass differences were calculated using a separation energy calculation. The numbers were small and then multiplied by a large number, meaning that the separation energy was sensitive to small changes in mass differences. Higher precision arithmetic was

used for this calculation i.e. the Decimal package in Python, to enable more precise values to be used.

5.10.2 Parameter Tuning

Parameter tuning is the name given to the process of tweaking the machine learning input parameters to improve performance. Each model has a unique set of input parameters that will affect the algorithm so there can be a lot of time spent if a trial-and-error approach is used. In some circumstances I was able to apply a tuning program that carried out the trial-and-error process automatically until an optimised set of input values were found.

6 Chapter 3: Experiments

6.1 Experiment 1: Predicting Mass Values

6.1.1 Overview

Many nuclei have had their masses successfully measured. Theoretical models tell us many more exist. Several theoretical models exist which all give slightly different answers for theoretical mass. Machine learning may enable us to predict masses without relying on a particular theoretical model, allowing for a more stable mass calculation, or a confirmation of a specific theoretical model. Experiment goal: Given Z and N is it possible to predict the mass of nuclei?

6.1.2 Data

Data Source: The data used was from (ChartofNuclides)
Features: Z and N were used as the features
Target variable: Mass

6.1.3 Method

Predicting mass is an unsupervised regression problem since the data is unlabelled. I selected several regression models, optimised their input parameters, and compared their results.
The models chosen were,

- Random forest.
- Gradient booster.
- SVM.
- Neural network.

6.1.4 Results

The mean absolute error was found for each model. The SVM model performed the best.

ML model	MAE
Random Forest Test MAE	0.6266648263151702
Gradient Boosting Test MAE	0.48021033981287115
SVM Test MAE	0.05012349441225703
Neural Network Test MAE	0.09059823108321653

Table 5: MAE for Models Predicting Mass Values

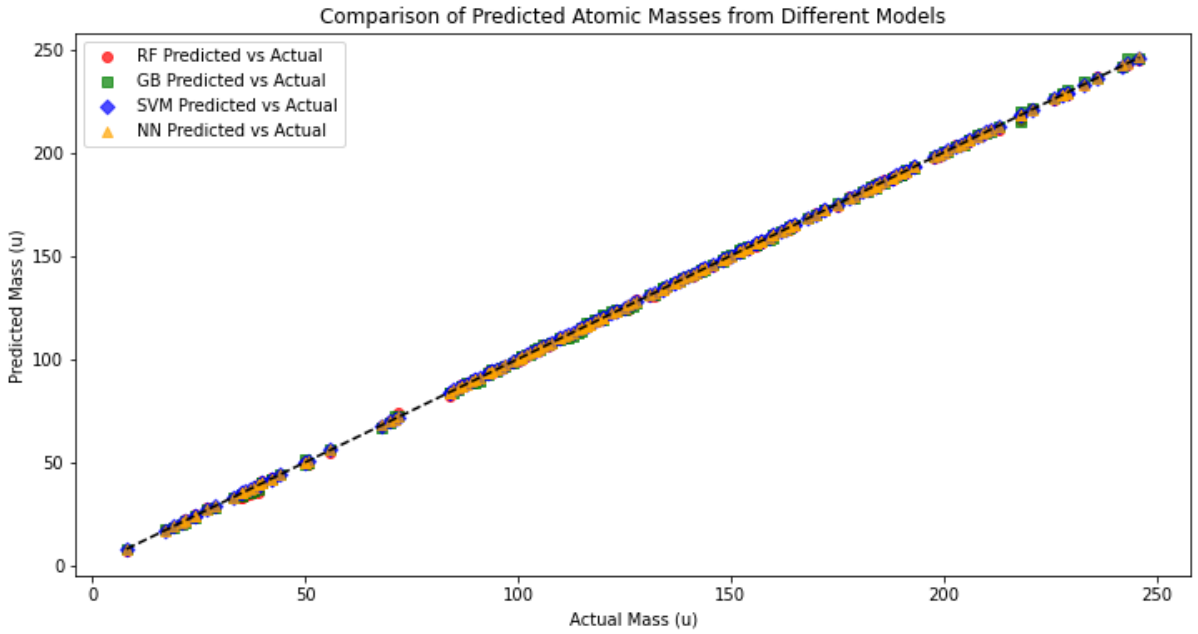


Figure 6: Actual mass overview

On first inspection the predicted values of mass appear to match well with the expected values but on closer inspection the values are not an exact match.

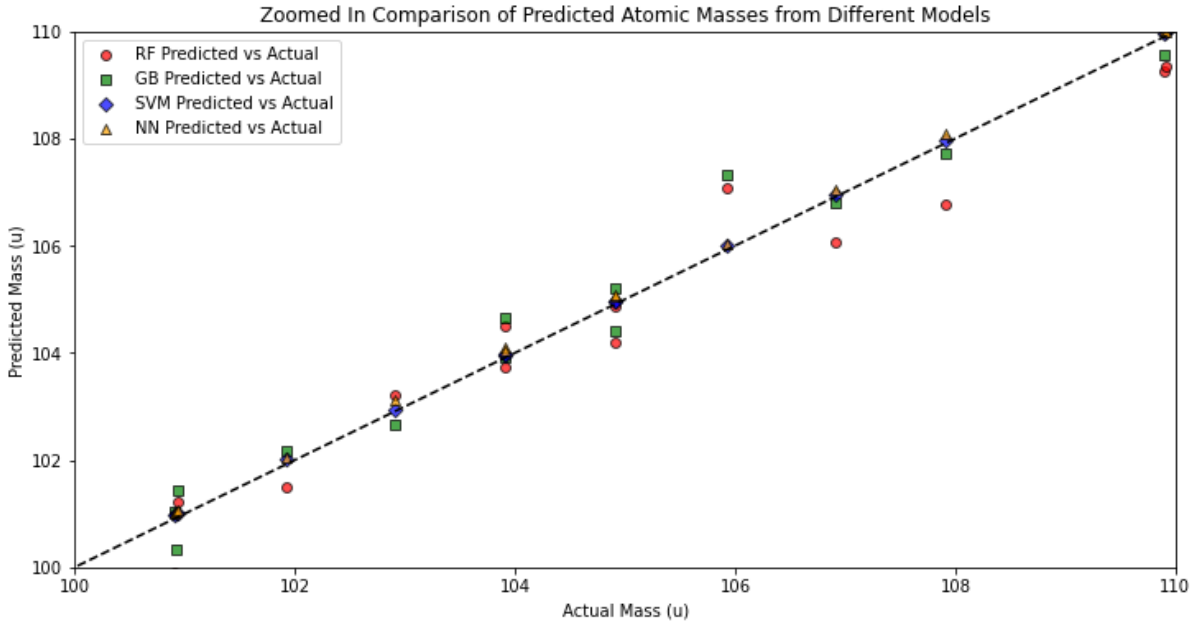


Figure 7: Actual mass zoomed in

It is clear from the MAE value and a visual inspection that the SVM model performs better than the others and it can predict masses with a reasonable degree of certainty. You can also see for the Random Forest model, the deviation from the actual values increases as we move to

heavier elements due to the fact there are less data points for heavier elements. Random forest appears to provide the least accurate results with the given data.

6.1.5 Conclusion

Masses of known nuclei have been experimentally measured to a high degree of accuracy. Machine learning models are of little use on this known data as they are only telling us what we already know. Where the model will be useful, will be for predicting masses of unknown nuclei. It is possible to use current physics model to predict masses, but this relies on previous physics knowledge and the selection of an appropriate model. The goal of this experiment is to predict masses without prior physics knowledge. This model can be used to predict the mass of a 'theoretical' isotope with some degree of confidence.

6.1.6 Further work

The calculated masses could be compared to existing model data as documented in (Theoretical description of nuclear masses, 2021)
To increase confidence more features could be added.

6.2 Experiment 2: Predicting Separation Energies

6.2.1 Overview

If we can predict the separation energy for a theoretical isotope, we should be able to predict at what particular N the separation energy is equal or less than 0 and therefore indicate when the drip line has been reached.

6.2.2 Data

Data Source: The data used was from (ChartofNuclides).
Features: Z, N and theoretical mass were used as the features.
Target variable: Separation energy.

6.2.3 Method

Predicting separation energy is an unsupervised regression problem since the data is unlabelled. I selected several regression models, optimised their input parameters, and compared their results.
The models chosen were,

- Random forest.

- Gradient booster.
- SVM.
- Neural network.

6.2.4 Results

The mean absolute error was found for each model. The neural network model performed the best, bust still quite poorly overly.

ML model	MAE
Random Forest Test MAE	1183.6098577455862
Gradient Boosting Test MAE	1203.0767211043876
SVM Test MAE	1314.3716228199137
Neural Network Test MAE	1165.7421416535942

Table 6: MAE values for models Predicting Separation Energies

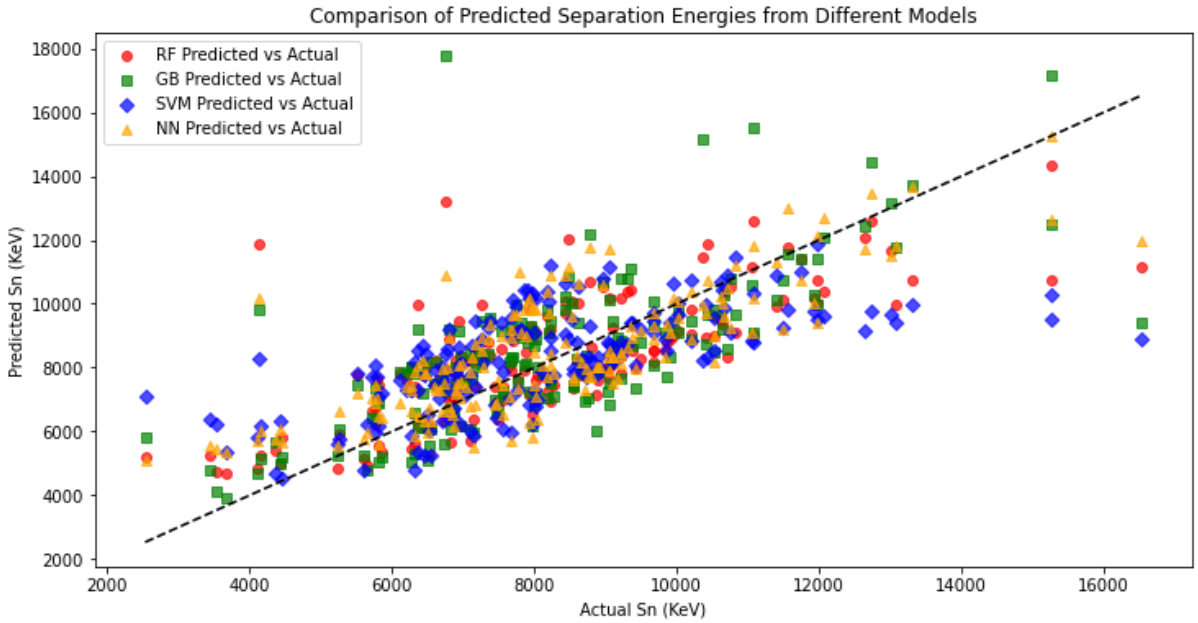


Figure 8: Actual S(n) versus predicted S(n)

Once a neural network model was identified as the best model, I developed a more detailed neural network model to improve the results. The model used was 'Sequential' from Tensorfow and Keras. Keras randomsearch tuner was used to optimise the hyper parameters.

The results can be found in the scatter plot in figure 9. This figure shows two distinct patterns in the scatter plot.

1. There is an obvious linear pattern following the expected values of Sn but some of the predicted values are scattered from the trend line, indicating the existence of prediction errors. There is a group of predicted values quite close to the black line indicting perfect predictions, this shows the model is performing better in the range of around 6000 to 12000 keV.
2. After around 12000 keV the linear pattern stops but the presence of outliers remain showing the model doesn't perform well predicting values for the heavier elements.

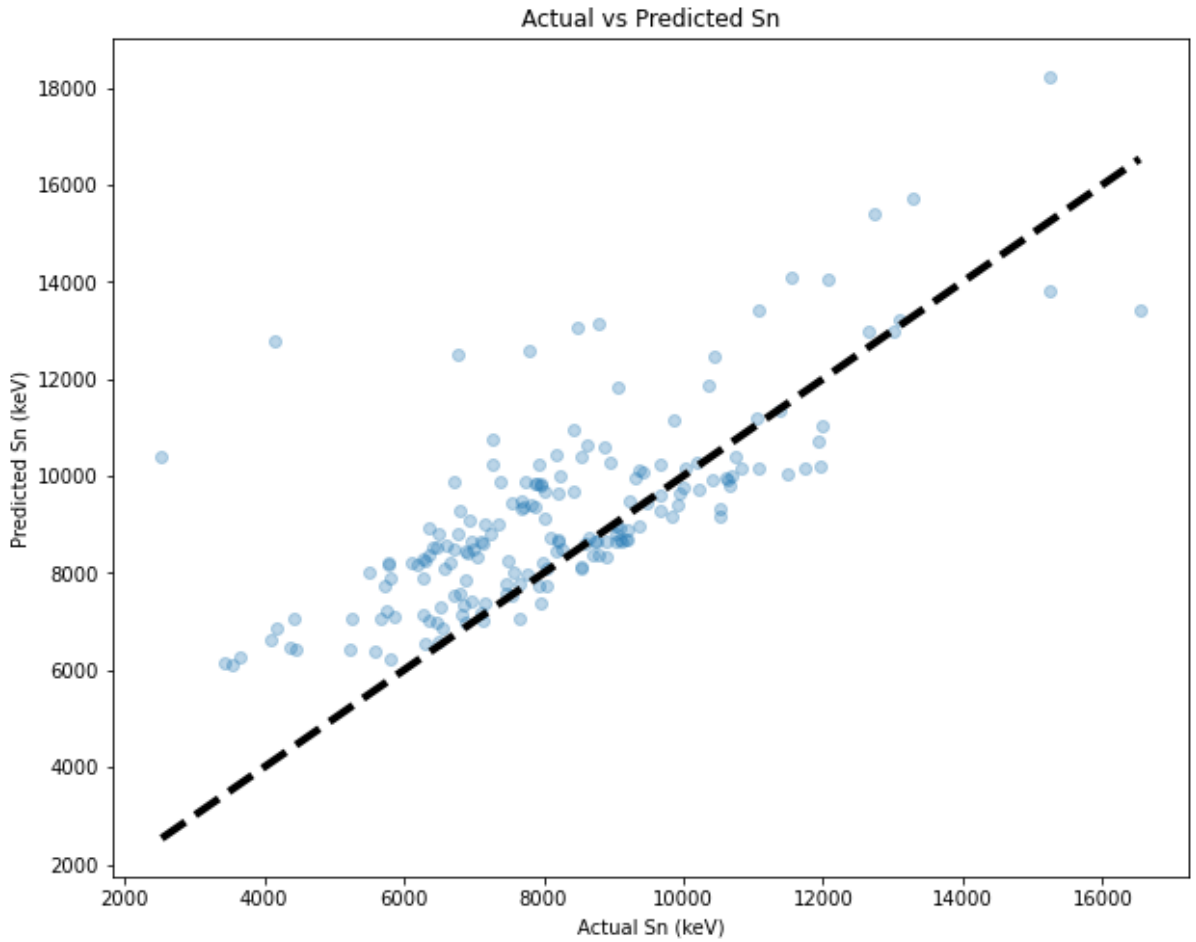


Figure 9: Actual S(n) versus predicted S(n) improved

6.2.5 Conclusion

The fact that the neural network model performed better shows that the relationship between N, Z, mass, and separation energy is non-linear, since this type of model is better at learning

non-linear relationships than the other models. In experiment 1, the relationship between Z, N and mass is linear which is why the SVM model performed well.

The optimised neural network model improved the performance slightly, but they were still not as accurate as hoped. By looking at which model performs best we can start to understand they types of relationship between parameters, i.e. linear / non-linear.

The poor performance at higher values of Sn could be due to the following reasons.
Data distribution. The training data may have more samples in the 6000 keV to 12000 keV range, leading to more accurate predictions within this range. The likely reason for poor prediction quality beyond 12000 keV is the scarcity of data at higher energy levels, which is due to the shorter life span of isotopes at those energies. There is often less recorded data at higher energy levels because these excited states decay rapidly, making them harder to detect and measure. This contributes to fewer data points in higher energy ranges, impacting the accuracy of predictions in machine learning models that rely on this data.

Complex physics. At higher energy levels, there might be intricate and less understood physical phenomena at play that the machine learning model isn't capturing. This could be due to the limited or skewed nature of the training dataset, which might not fully represent these complex interactions. Since the model can only identify patterns based on the data it has been trained on, it struggles to accurately predict outcomes in regions where the underlying physics behaves differently or where it has insufficient data. Consequently, it fails to generalise well to these high-energy scenarios.

6.2.6 Further work

More features should be added to the dataset to understand what influences the relationship between Z, N, mass, and separation energy. Of course, we could speculate according to physics models but if we are trying not to influence the models with prior knowledge, features should be added systematically until an improved result is found.

The experiment should be split into two further experiments. One at energies below 12000 keV and one for energies above 12000 keV. This may result in more accurate results for the values above 12000 keV since those data points are more likely to be experiencing the same physics phenomena.

6.3 Experiment 3: Identifying Nuclear Shell Closures through Clustering of Separation Energies

6.3.1 Overview

In nuclear physics, understanding the structure and stability of atomic nuclei is paramount. One critical aspect of nuclear structure is identifying the closure points of nuclear shells, which significantly impact the chemical and physical properties of elements. Traditional methods of exploring these shell closures rely on experimental measurements and theoretical calculations, which can be resource-intensive and computationally demanding.

This experiment introduces a novel approach using machine learning to uncover patterns in nuclear separation energies, specifically targeting the identification of last proton shell closures. By applying clustering algorithms to datasets of separation energies, the last proton in each cluster should represent the final proton in the shell. This method not only offers a potentially faster and less resource-heavy avenue for identifying shell closures but also provides a unique lens through which nuclear stability can be examined.
The primary objective of this experiment is to identify the last proton and neutron in each nuclear shell by applying clustering algorithms to separation energy data. This study aims to evaluate two key aspects: first, whether distinct nuclear shells can be accurately delineated through this method; and second if the clustering can reliably pinpoint shell closures.

6.3.2 Data

Separation energies for all isotopes from www-nds.iaea.org (ChartofNuclides)

A	Elt	Z	S(n)	-	S(p)	-
2	H	1	2224.57	0.0004	2224.5662	0.0004
3	H	1	6257.23	0.0004	*	*
4	He	2	20577.6	0.0005	19813.8661	0.0002
5	Li	3	21715.6	217.9449	-1964.9999	50.0000
6	He	2	1710.46	20.0001	22589.3230	89.4427
6	Li	3	5663.32	50.0000	4433.3246	20.0000
7	Li	3	7251.09	0.0045	9973.9616	0.0531
7	Be	4	10677.4	5.4482	5606.8539	0.0709

Figure 10: Small section of raw, downloaded data

6.3.2.1 Exploratory data analysis; investigating the separation data.

To familiarise myself with the data I performed an exploratory analysis of the data. It is good practice to do this with data so you can start to get an intuitive feel for any patterns or relationships that might guide any future designs regarding experiments.

6.3.2.2 Investigating the separation data

This data analysis is for the Sn1 data.

The data has 2112 entries and 18 columns.

The unique Z values are 1. 2. 3. 4. 5. 6. 7. 8. 9. 11. 12. 13. 14. 15. 16. 17. 18. 19.
21. 22. 23. 24. 25. 26. 27. 28. 29. 31. 32. 33. 34. 35. 36. 37. 38. 39.
41. 42. 43. 44. 45. 46. 47. 48. 49. 52. 51. 54. 53. 55. 56. 57. 58. 59.
61. 62. 63. 64. 65. 66. 67. 68. 69. 71. 72. 73. 74. 75. 76. 77. 78. 79.
81. 82. 83. 84. 85. 86. 87. 88. 89. 91. 92. 93. 94. 96. 95. 97. 98. 99.

The unique N values are: 1. 2. 4. 3. 6. 5. 8. 7. 10. 9. 12. 11. 14. 13.
15. 16. 18. 17. 20. 19. 21. 22. 23. 24. 25. 26. 27. 28.
29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42.
43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56.
57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70.
71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84.
85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98.
99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112.
113. 114. 115. 117. 116. 118. 119. 120. 121. 122. 123. 124. 125. 126.
127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140.
141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154.
155. 156.

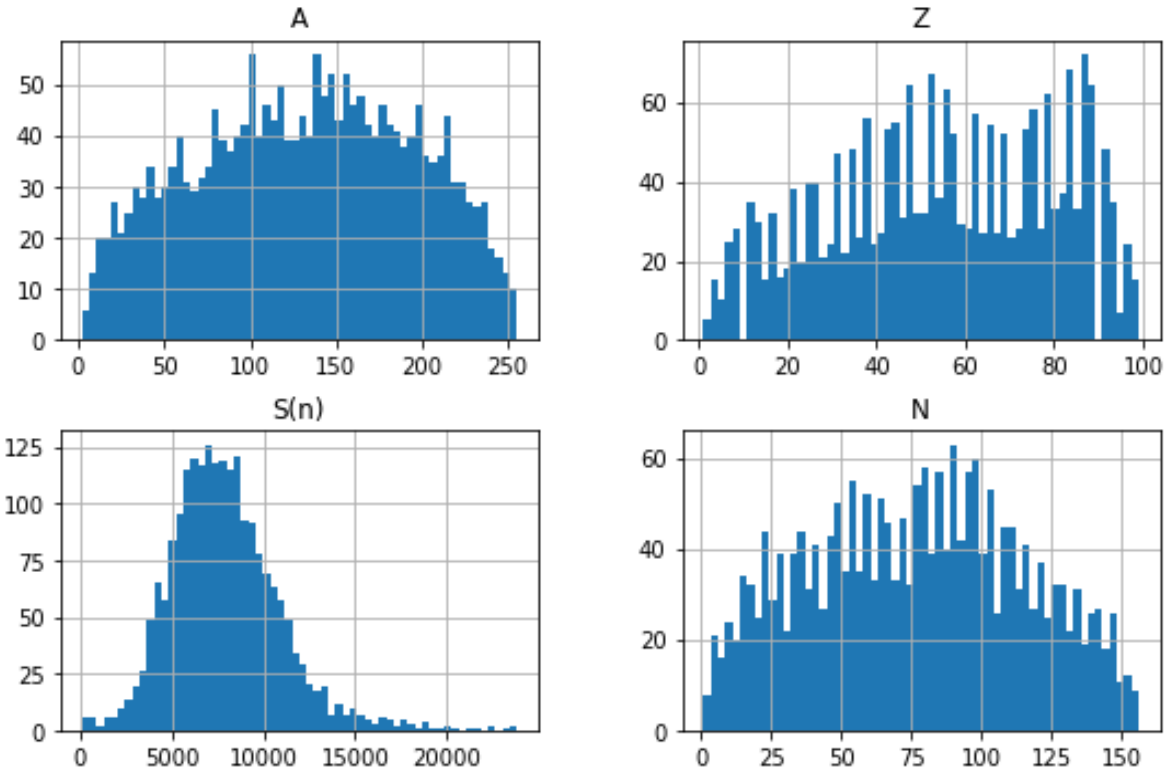


*Figure 11: Histogram for each numeric input variable**

*Please note, the histogram is only intended to give an idea of the distribution due to the bin sizes.

A (Mass Number): This histogram shows a roughly symmetric distribution of mass numbers with a peak somewhere around 100-120. This implies a higher frequency of stable isotopes in this mass range, but the reality is stable isotopes are not more abundant within a particular mass range, they are distributed widely across a range of mass values. This shows that the data is simply a reflection of the isotopes that have been studied.

Z (Atomic Number): The distribution for atomic numbers seems to be multi-modal, with two distinct peaks. One around the high 80's (88, 89) and one around 50. The high 80s represent highly radioactive elements which is an active area of research due to the application of radioactive isotopes in areas such as medicine and energy. Radium, discovered by the Curies, has a historical significance in the study of radioactivity and chemistry. This historical interest has led to extensive documentation and data collection. Z = 50 is Tin and has the highest number of stable isotopes. For this reason, it will be easier to collect an abundance of data on it.

S(n) (Neutron Separation Energy): This distribution appears right skewed, meaning there are a few isotopes with higher neutron separation energies that form a tail on the right side. This suggests that as the separation energy increases, fewer isotopes will be found.

N (Neutron Number): The distribution appears to have peaks around N = 21, 28, 50 and 82 which correspond with the magic numbers. This would mean that the isotopes where N equals a magic number would be studied more due to their stability and interest.

To summarise, it is clear the data is skewed in favour of the elements and isotopes that are more abundant due to stability, easy to measure or have significant cause to study them. It isn't a complete set of data. This must be considered when performing analysis later.

Investigating Sn against N.

In the next part of the data exploration neutron separation energy was plotted against the neutron number on for a particular element, in this example, calcium.

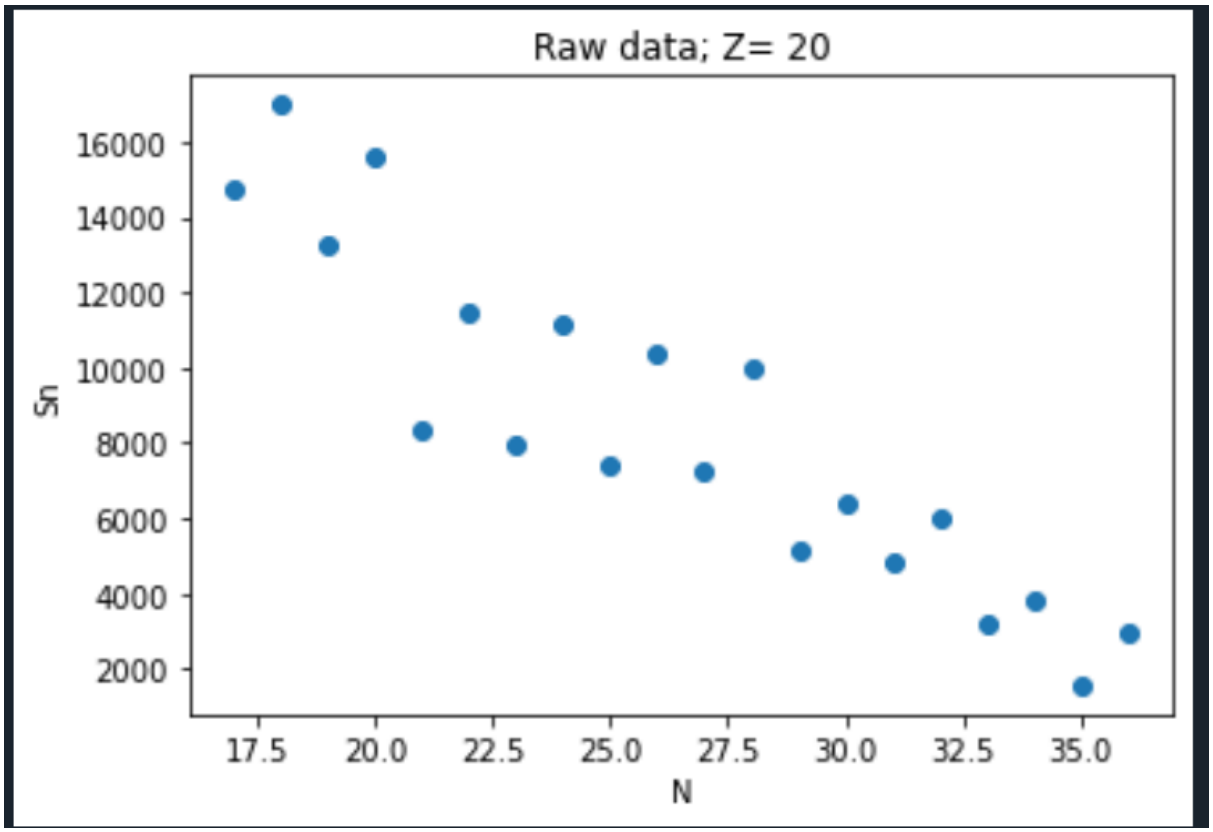


Figure 12: Sn against N for Calcium

The diagram above shows the value of N increasing as the value of Sn decreases. Three clear groups or clusters can be seen. They coincide with the theory of the shell model.

This is where the idea for the first experiment came from.
Could machine learning show shell structure?

The diagram above illustrates the relationship where the value of N (neutron number) increases as the value of Sn (neutron separation energy) decreases. Three distinct groups or clusters are evident, aligning with the theoretical predictions of the shell model. This observation sparked the initial concept for the experiment with the question; could machine learning effectively reveal the underlying shell structure by identifying these grouped patterns?

6.3.3 Method

To answer the question above the following activities need to take place.

1. From visual analysis of the separation data (see figure above) it could be seen that the data can be separated into groups. In the diagram above three distinct groups can be seen. The first step is to find a ML model than can group the data.
2. The second step is to use regression on the group of data closest to Sn = 0 (bottom of the diagram) to find the line of best fit.
3. Predict where the separation energy = 0 by extrapolating the line of best fit to make a prediction outside of the given data set.

The first step is a clustering problem. Since the data has no clustering information within it, an unsupervised clustering algorithm will be used.
A comparison between the following models was made,

- K-means.
- DBSCAN.
- Agglomerative.
- Mean Shift.

The clustering model ‘K-Means’ was selected.

Model limitations: The model was set up with a cluster value of 3, this meant it would only work with 3 or more data points in a set so elements such as hydrogen and any other with very small data sets were excluded. This was later changed to 2 clusters which gave much better results.

6.3.4 Results

The shell closure data was split into two categories, major shell closures and minor shell closures.

Major shell closures are N = 2, 8, 20, 28, 50, 82, 126

Minor shell closures are 6, 14, 16, 32, 38, 40, 58, 64, 68, 70, 92, 100, 106, 110, 112, 136, 142, 154, 162, 164, 168.

Below is a selection of clustering results.

It can be seen that the K-Means method is sufficient at identifying clusters when two or three are present. When there are only two clusters present there is sometimes a small amount of overlap between the clusters. This will be a source of error in the final conclusion. Fortunately, the dataset is large enough to put up with the existence of some errors.

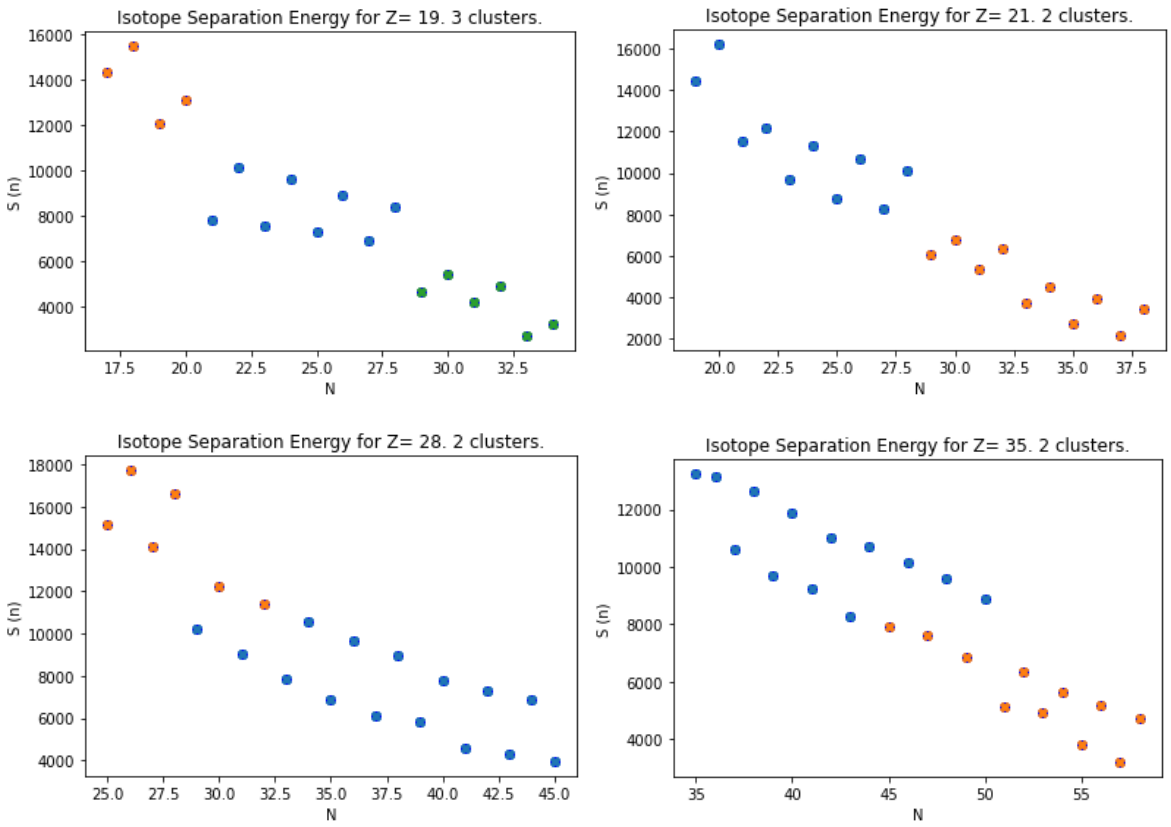


Figure 13: Results from clustering algorithm showing clusters

Below is a selection of plots showing the final nucleon being correctly identified. Below shows major shell closure at 8 and minor closure at 14. 20 wasn't identified because it belonged to the third cluster which was omitted from the analysis. Interestingly, the first plot shows two

clusters, but the cluster algorithm identified 3 clusters and correctly identified 14 as a minor shell closure. The second plot correctly identified 20 and 28.

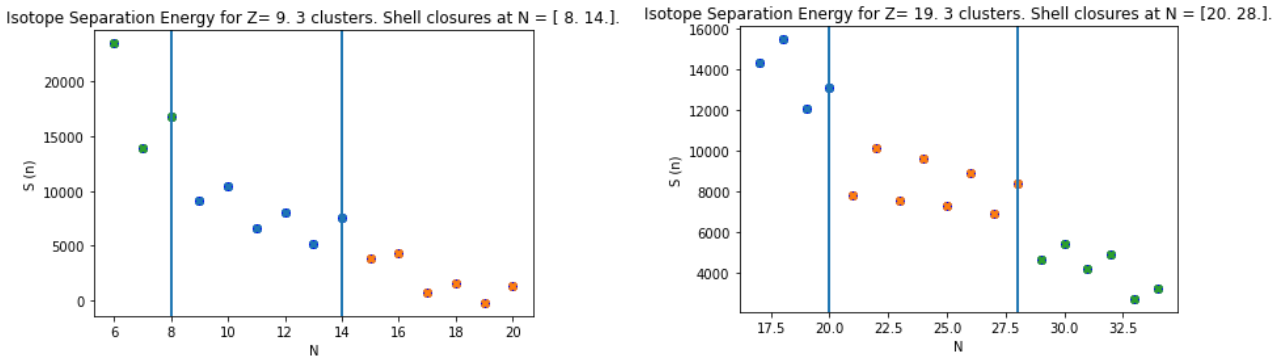


Figure 14: Shell closures

Below shows a histogram plot of the final shell closures. This data is biased because more data exists in the centre of the data set due to the nature of the data.

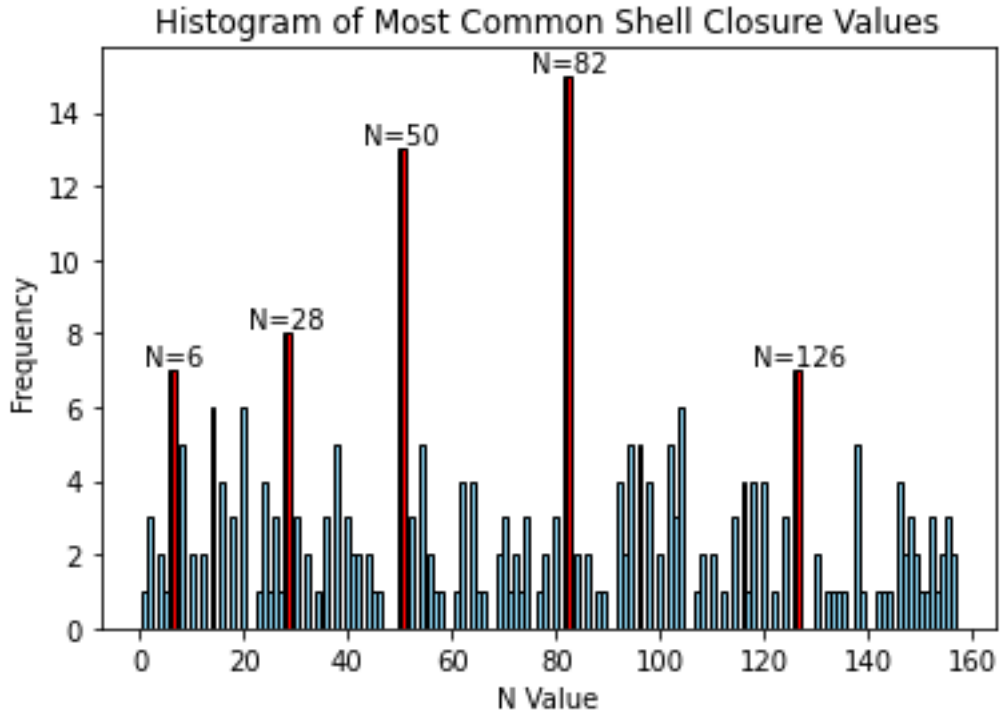
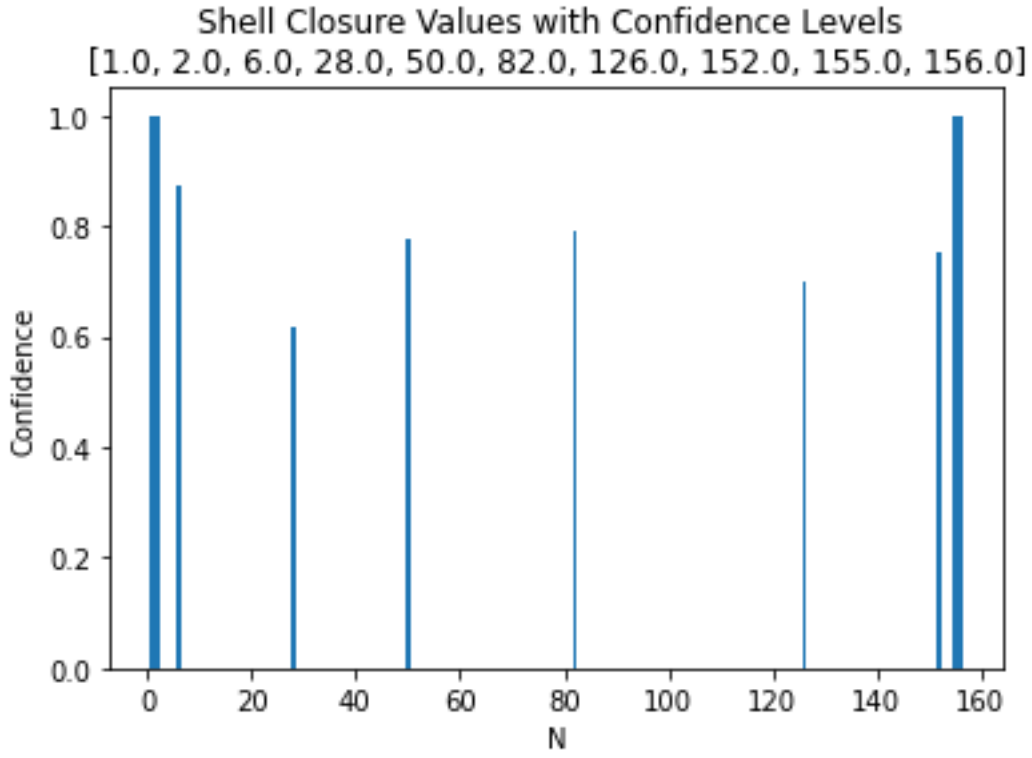


Figure 15: Histogram of shell closures

Given the inherent skewness present in the dataset, the use of confidence levels was incorporated to normalise the data, ensuring a more balanced and meaningful analysis. The normalisation process assessed each shell closure's frequency relative to its potential frequency of occurrence. The adjusted criteria meant that only shell closure values with a confidence level exceeding 0.6 are deemed significant. This threshold indicates that a given value is observed in at least 60% of the instances where it could theoretically appear,

according to the model. Applying this stringent standard, we've identified a set of shell closure values was identified that consistently emerge with high reliability. 0.6 was chosen through trial and error. Any higher and most of the ‘magic numbers’ were eliminated. Any lower and too much ‘noise’ was included.



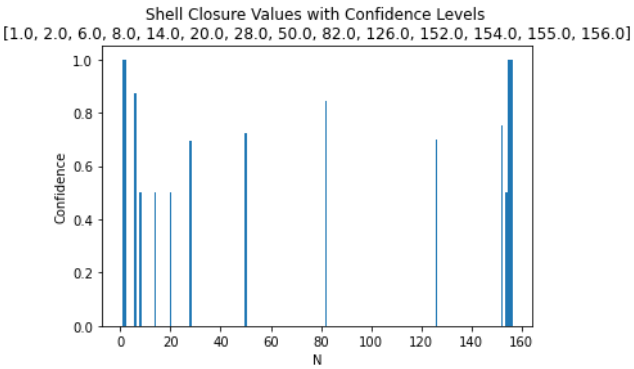


Figure 16: Confidence level of 50%

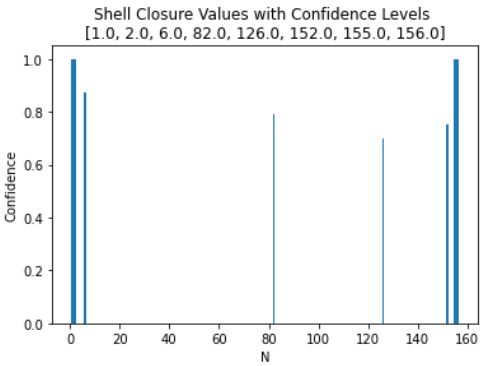


Figure 17: Confidence level of 70%

Taking a closer look at the outliers I noticed a high confidence level for N = 1, 2, 152, 155 and 156. These are not traditionally shell closures (apart from 2) so further investigation was carried out. I suspected that there were very few data points in the dataset for these. If this was the case it wouldn't take many results to give these a high confidence vote.

My suspicions were correct.

N	Data points
1	1
2	3
152	4
155	3
156	2

Table 7: Table of outliers

I made the decision to discard any datasets with 4 or less data points.

Setting the confidence level to give the best results.

To recap, the confidence level quantitatively measures the proportion of times a particular value appears relative to the number of times it could potentially appear. Essentially, it assesses how frequently an observed value occurs in relation to its expected frequency, providing a standardised indicator of its significance or reliability in the data.

6.3.5 Conclusion

The shell closures of 20, 28, 50 and 126 have been calculated with a high degree of confidence.

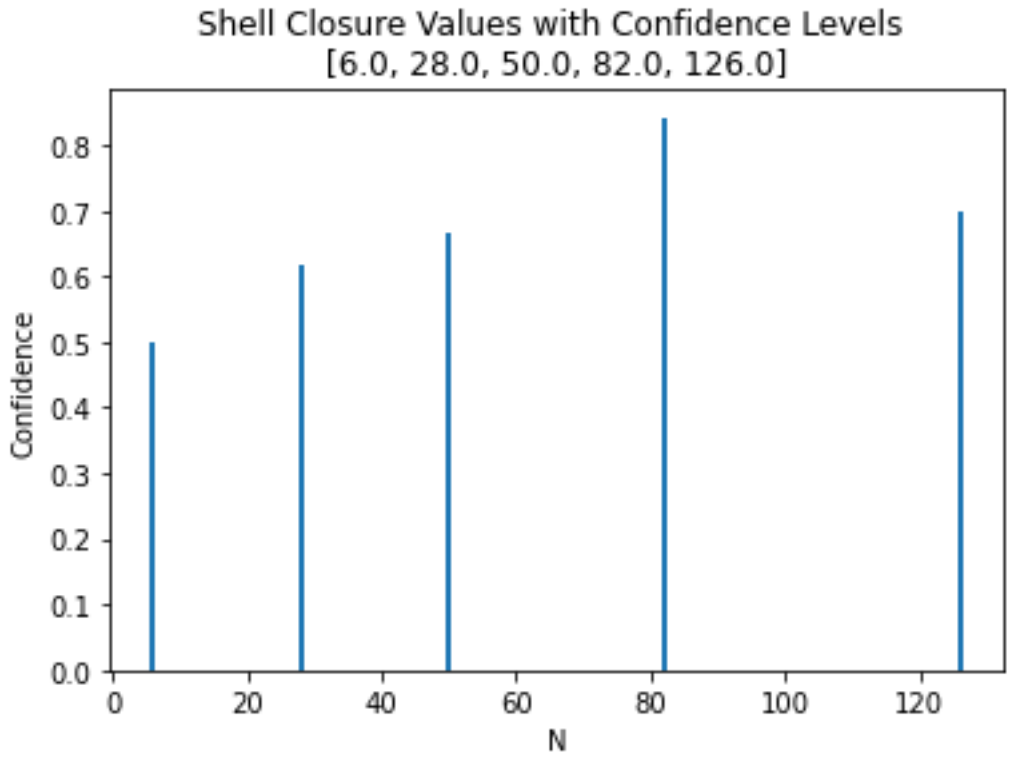


Figure 18: Shell closures with added higher confidence

These numbers align with magic numbers as shown in figure 18.

Reducing the confidence levels to 0.5 enabled the inclusion of minor shell closures of 6, 14 and the major shell of 20 as shown in figure 19.

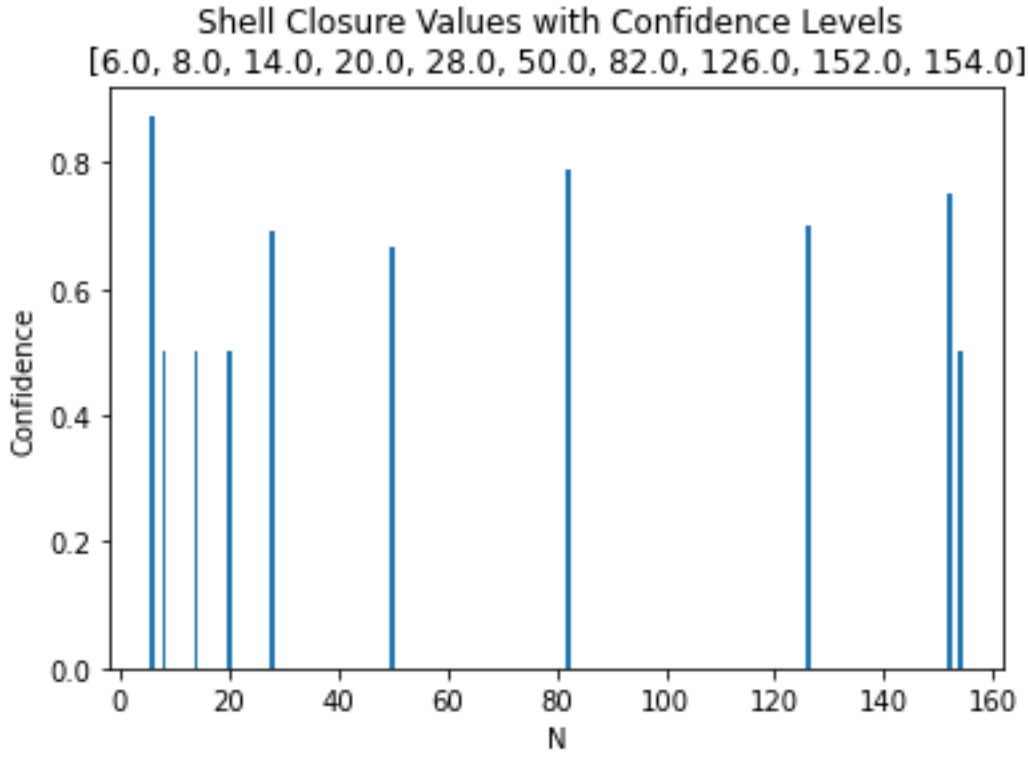


Figure 19: Shell closures with added lower confidence

6.3.6 Further analysis: Regression

Any plot of Sn V N, see figures 20 and 21, will show the value of N increasing as the value of Sn decreases.

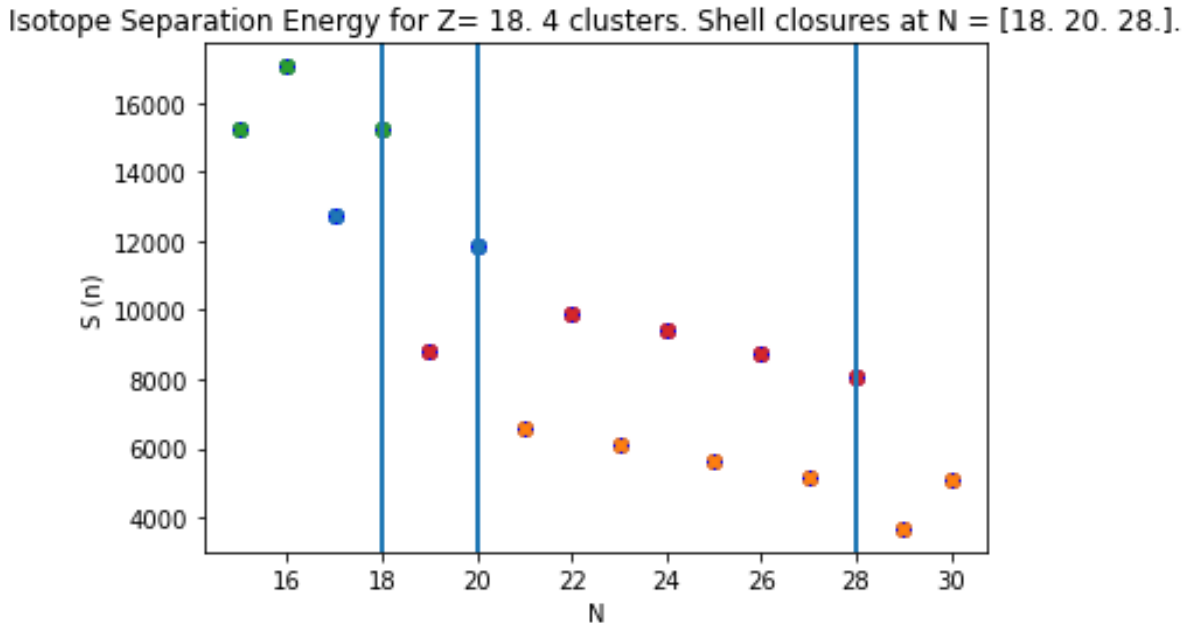


Figure 20:Multiple shell closures for Z = 18

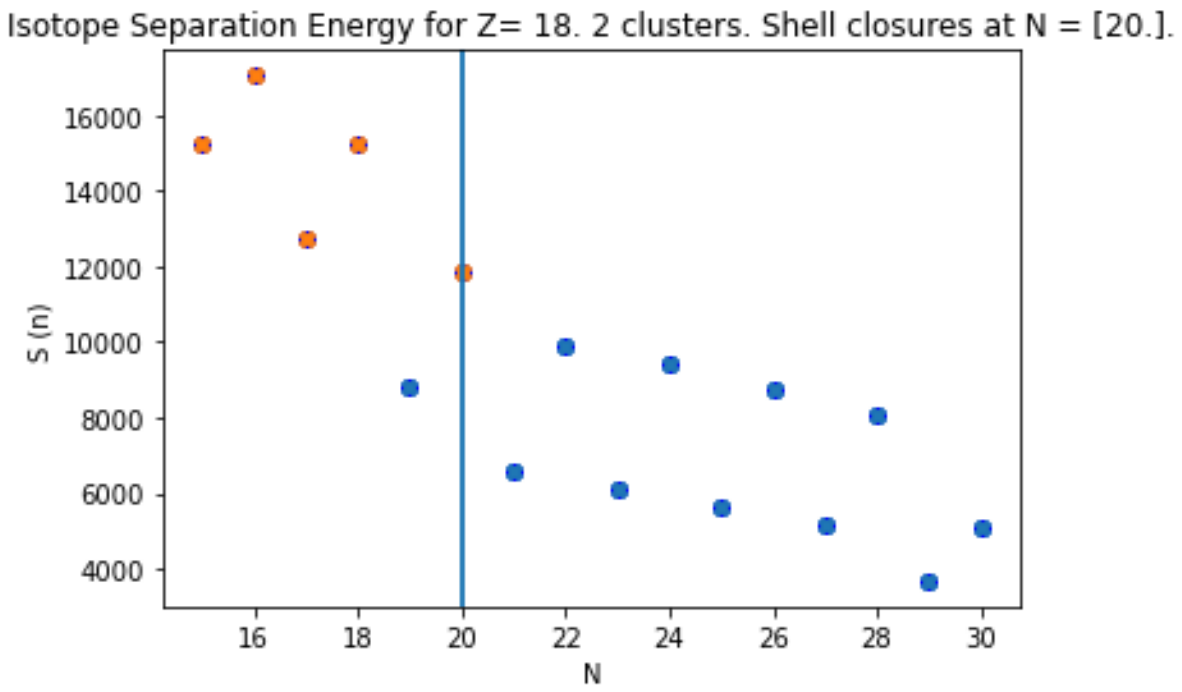


Figure 21: Shell closure for Z = 18

As Sn decreases the stability of the isotope decreases and as Sn reaches zero the isotope will decay by emitting a neutron. The value of N, beyond which no more isotopes can be found is called the drip line. Investigating the Sn V N diagram gave rise to this analysis that aims to answer the question; can we predict the value of N at which Sn is equal or less than 0 i.e. can we predict the drip line?

6.3.6.1 Method

In order to answer the question above the following activities need to take place.

1. Using the data on the clusters, find the cluster closest to the y (Sn) = 0 line.
2. Using regression on this data, find the line of best fit.
3. Predict the final value of N, just before where the separation energy = 0 occurs by extrapolating the line of best fit to make a prediction outside of the given data set.

6.3.6.2 Results

I tried fitting a straight line to the final cluster, but this didn't always show a best fit. The best fit line looked like it should be slightly curved. I tried with many orders of polynomial curve fitting but with no success. I decided the linear extrapolation method, while not perfect, was

the best solution for this experiment. Using linear regression analysis on K-Means clustering data gave the following results.

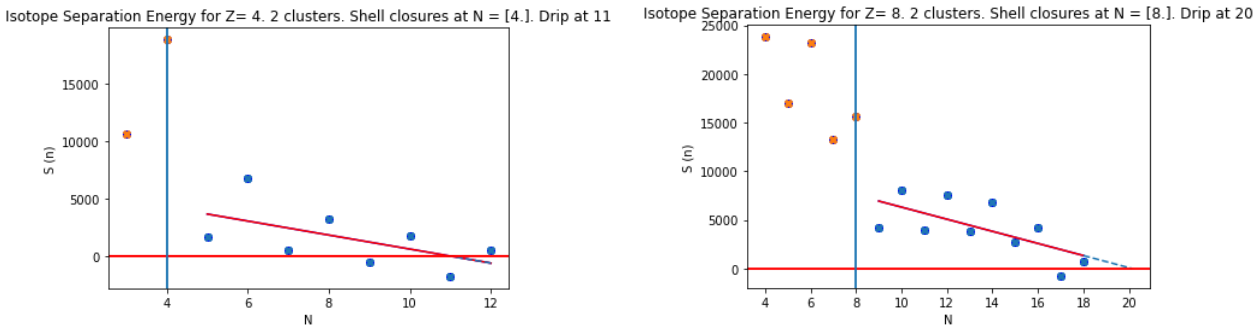


Figure 22: Finding the drip line using two clusters

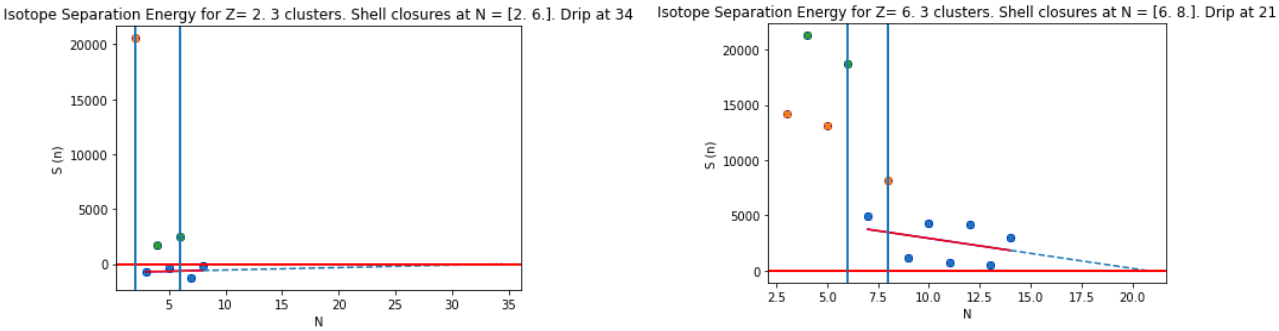


Figure 23:Finding the drip line using three clusters

Comparison data taken from (ChartofNuclides)

Z	Species	N (observed)	N (predicted 2 clusters)	N (predicted 3 clusters)
1	^{5}H	4	3	7
2	^{10}He	8	8	34
3	^{12}Li	9	9	9
4	^{14}Be	10	11	11
5	^{21}B	16	13	13
6	^{22}C	16	17	21
7	^{24}N	17	18	32
8	^{26}O	18	20	20
9	^{31}F	22	20	20
10	^{34}Ne	24	No Data	No Data

Table 8: Predicted N Values Using Sn

6.3.6.3 Regression conclusion

Using two clusters the drip line was predicted exactly in 2 out of the 9 cases above and to within +- 1 in 6 out 9 cases, or in 70% of cases.
Using three clusters only 1 value was predicted exactly and 2 within -+ 1. This was due to the small amount of data points contained within the final cluster when the data was split into three clusters. This analysis proved that it is important to have as much data as possible for regression. Too little data gives poor results because there is not enough data to fit a line to.

This shows that splitting the data into two clusters gave the best results when trying to predict the drip line position.

6.3.7 Further analysis: P drip line

6.3.7.1 Method

The same process was applied to proton data. The proton drip line is more well observed than the neutron drip line.

An initial look at the plot for S(p) versus N showed an interesting feature in the data. Where N = Z there was a significant step up in Sn. This indicated the protons in nuclei were more stable when Z = N. The other significant observation was the lack of clusters of data as could be easily seen in the Sn V N plots. The existence of clusters showed a resemblance to the shell structure, and it highlighted the nuclei that were more stable, giving rise to the 'magic numbers'. If we are to believe the shell model, then the lack of clusters in the S(p) data shows that there are other forces at play resulting in less stable, proton heavy nuclei.

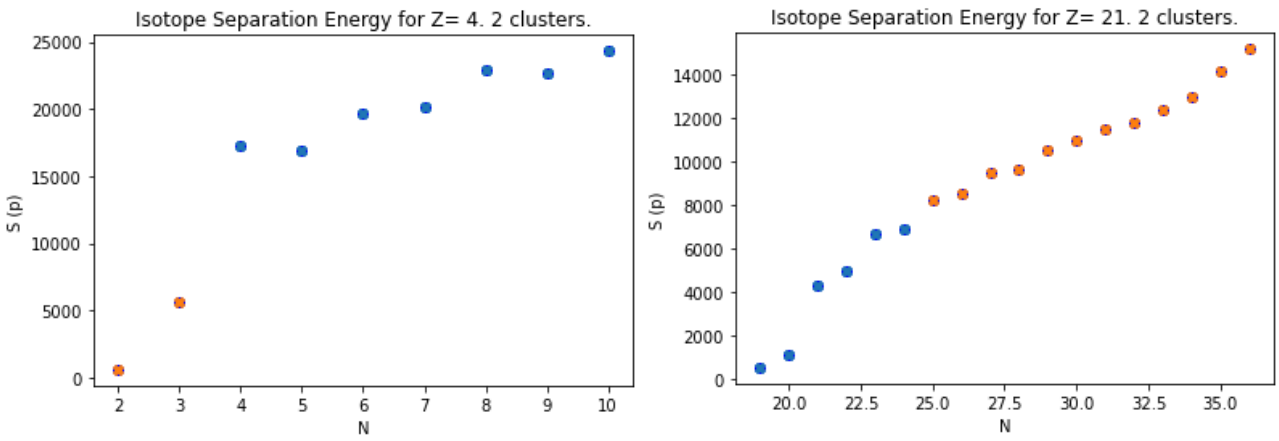


Figure 24: Sn Clusters

Due to the smoother appearance of the Sn data, it wasn't possible to predict magic numbers or shell closures. However, clusters could still be identified, and regression used as before to predict the proton drip line.

6.3.7.2 Results

Note: Regression was extended to two clusters if the cluster data set contained too few data points (< 3).

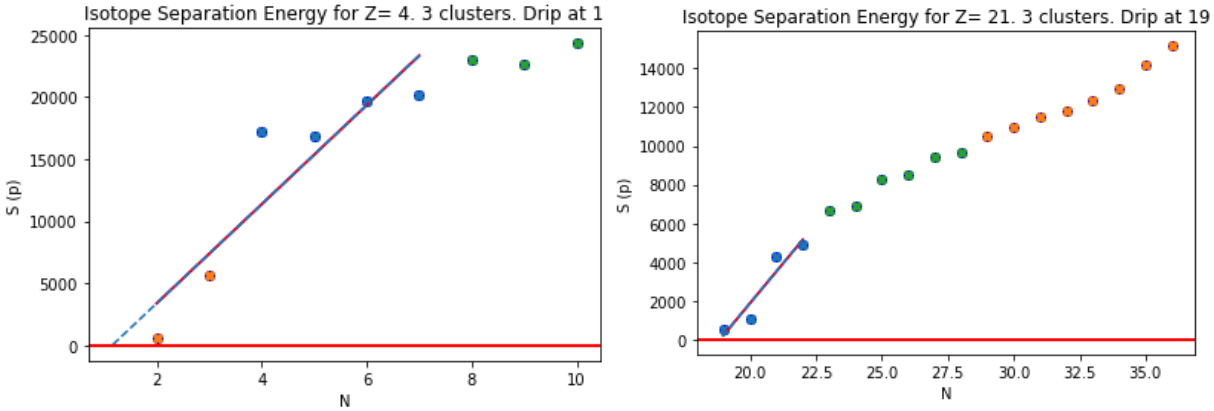


Figure 25: Predicting proton drip line

Z	Species	N (observed)	N (predicted 2 clusters)	N (predicted 3 clusters)
3	5Li	2	0	2
4	^{6}Be	2	0	1
5	^{7}B	2	2	3
6	^{8}C	2	3	3
7	^{11}N	4	5	5
8	^{12}O	4	5	4
9	^{16}F	7	8	7
10	^{16}Ne	5	No Data	No Data

Table 9: Predicted N Values using Sp

6.3.7.3 P-drip line conclusion

The results for the first few nuclei show that it's much harder to perform regression since the clustering is more difficult with this smooth data. Moving to three clusters helped with the

predictions enabling the value to be predicted within +- 1. It is clear there are not three clusters in this data, moving to three clusters is just a way to influence the regression calculation to lean towards the data points closer to the y = 0 line. I don't think this is the right way to do it but I did it to highlight the fact that machine learning models can find it hard to predict patterns in data when the pattern is very subtle. In this case, the data points near y = 0 are very important for predicting the drip line but because there are few of them, they are contained within a cluster that they don't really belong to. They should be in their own cluster. Even if they could be identified in their own cluster there are too few data points to make a meaningful regression analysis.

6.3.8 Conclusion

Clustering was effective in identifying the patterns in the S(n) data because it contained easy to identify patterns. Shell closures for major and minor shells could be identified. It was harder to find patterns in the S(p) data was more difficult because the data was smoother.

6.3.9 Further work

Using clustering on linear data was the best algorithm from the many I tested but if I were to continue this work, I would try other machine learning models. I only tried a small set of clustering models but many more exist.

Regression analysis is only as good as the cluster definitions. I decided on straight line analysis after many polynomial experiments. Further polynomial experiments may reveal a better fitted curve than the linear as the linear pattern was limited when used on smaller datasets.

It was difficult to perform clustering on data sets as small as this. Clustering is better suited to larger datasets.

6.4 Experiment 4: Predicting Stability using Half Life Parameter

6.4.1 Overview

Understanding the stability of isotopes is fundamental in nuclear physics. Stable isotopes are those that do not decay over time, while unstable (radioactive) isotopes undergo radioactive decay, emitting radiation. Predicting stability helps in studying nuclear structure, decay modes, and nuclear reactions.

I wanted to see if there was an underlying relationship between N, Z and stability since the line of stability is fairly linear. Could it be predicted by machine learning models?

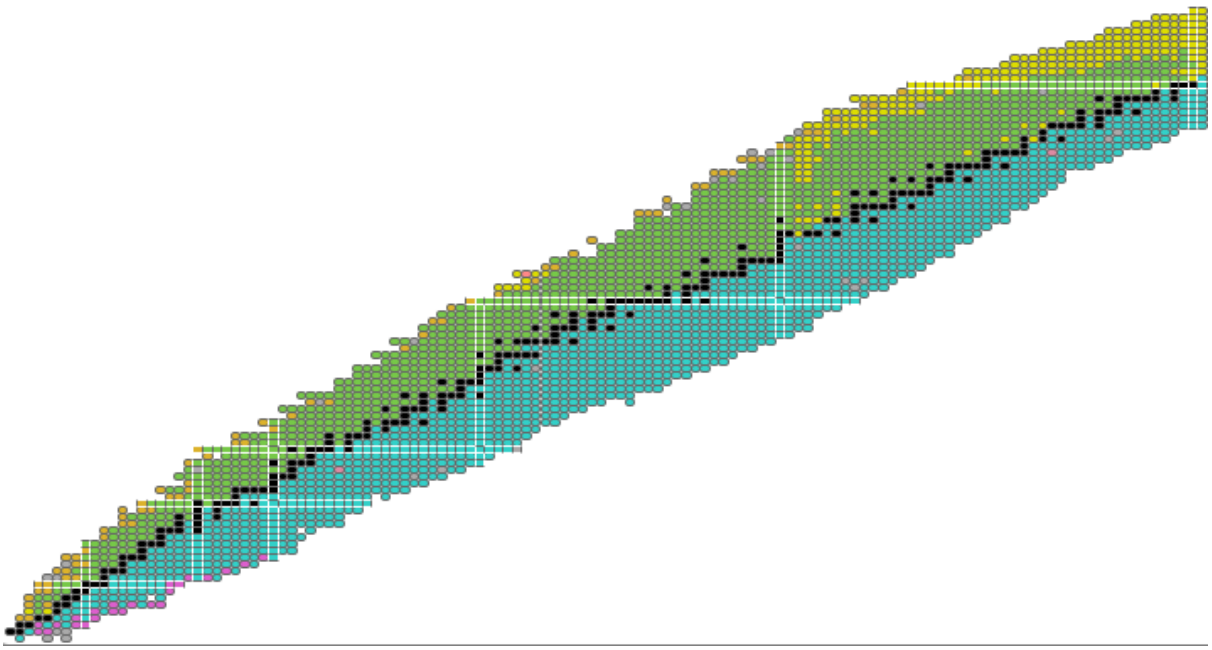


Figure 26: Chart of nuclides showing stable isotopes in black

6.4.2 Data

The ground state half life values were taken from the Chart of Nuclides (ChartofNuclides) and those which said ‘stable’ were marked with a 0 and those that were not stable were marked with a 1.

Index	z	n	symbol	idx	nergy_shi	energy	unc_e	ripl_shift	jp	jp_order	half_life
0	1	0	H	0		0			1/2+	1	STABLE

Figure 27: Raw data showing 'stable'

This was translated into a table of stability. There are many more unstable than stable isotopes. This gave a class imbalance.

Z	N	Stability
1	0	0
1	1	0
1	2	1
1	3	1
1	4	1
1	5	1
1	6	1
2	1	0
2	2	0

Figure 28: Table of stability

Z and N were used as the features. Stability was used as the target variable.

6.4.3 Method

We were only looking for a zero or one indicating stable or not, so this problem was a supervised binary classification model.
Several models were tested and the random forest model performed the best.

6.4.4 Results

The results were,
Correctly Predicted Stable Isotopes (predicted 0s): 22 (34.92%)
Correctly Predicted Unstable Isotopes (predicted 1s): 756 (97.30%)
The model found it much easier to predict unstable isotopes due to the fact they were more present in the data. This caused class imbalance as already mentioned.
To address the class imbalance, oversampling was applied. However, this approach did not significantly improve the results. I believe this outcome was due to the inherent imbalance in the data, which was a natural characteristic rather than an artifact of data processing.

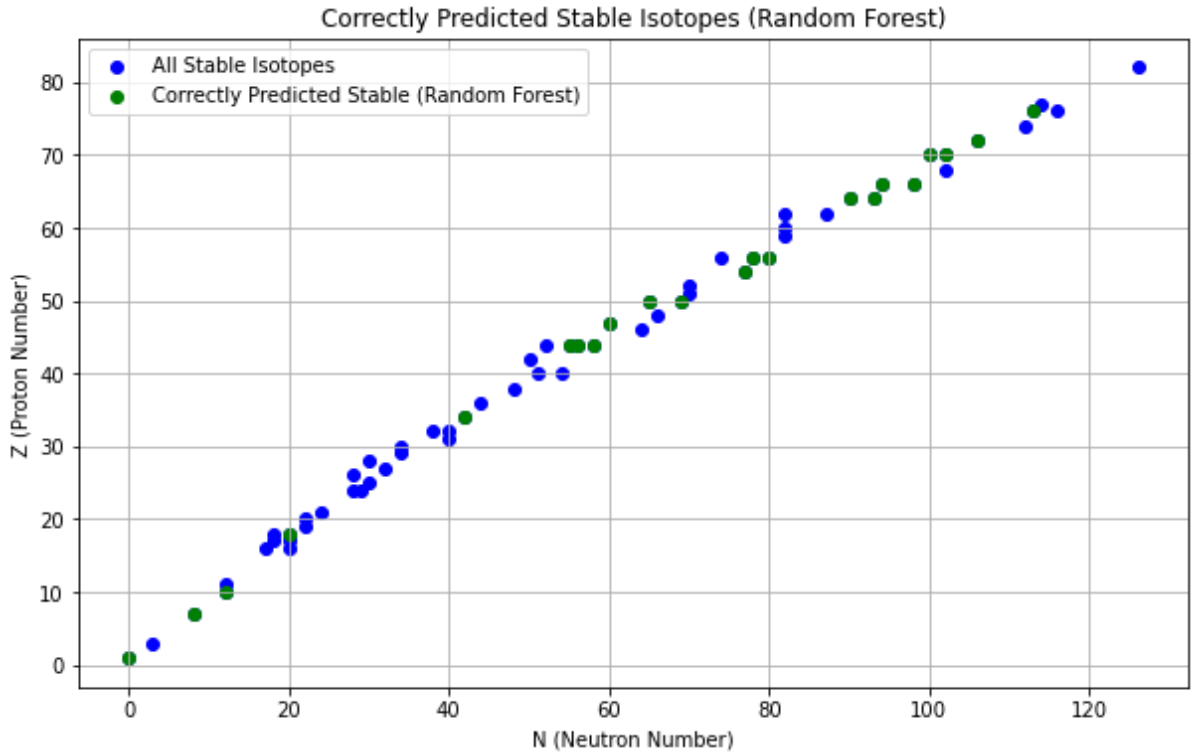


Figure 29: Correctly predicted stable isotopes

Feature addition.

I also added in another feature ‘magnetic_dipole’.

There wasn’t a magnetic dipole value for all values of Z and N so after removing all values with no magnetic dipole I was left with a much smaller data set.

171 unstable and 24 stable isotopes.

This meant it was still going to be difficult to predict stable isotopes.

The random forest could only predict one isotope correctly.

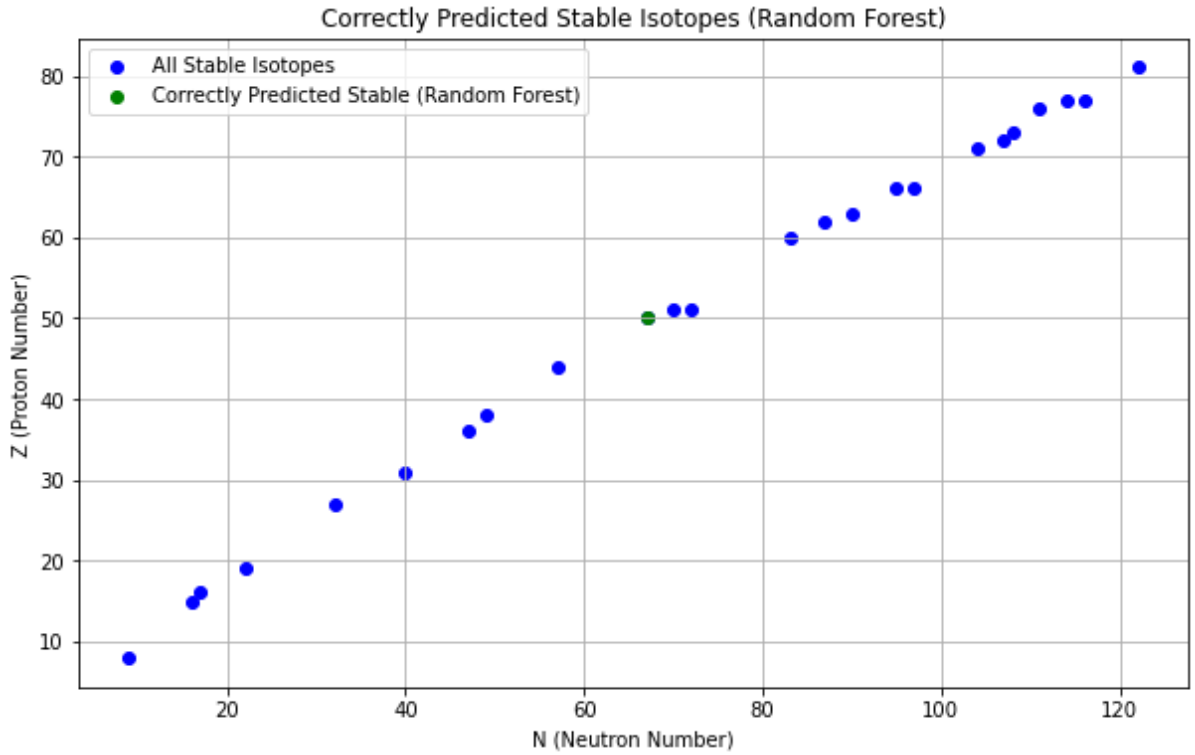


Figure 30: Predicting stable isotope with the inclusion of magnetic dipole

6.4.5 Conclusion

In this experiment we can see the importance of using a balanced data set. Even with the addition of balancing techniques such as oversampling, and the addition of another feature the data didn't improve. In fact, it deteriorated because the data set was reduced.

With this particular data set, the more features I add the smaller the data set becomes because it is not complete and consists of blank spaces, special characters and other characters that need to be removed. A large data set is important to give the best results.

6.4.6 Further work

More work needs to be done on how to balance data when there is a natural variance in the data. I don't think it's right to just remove some data to force it to be balanced. Many of the balancing techniques revolve around creating synthetic (fake) data. I think this would be a good area for further research. With more time I could introduce more features without reducing the dataset, while trying to balance out the stable data. This would take a lot of trial and error, but I am confident a solution on how to best balance data (for this type of naturally unbalanced dataset) could be found.

6.5 Experiment 5: Predicting Stability using Energy Level Densities

6.5.1 Overview

Energy levels inside a nucleus refer to the quantised energy states occupied by nucleons (protons and neutrons) within the nucleus of an atom. Nuclei can undergo excitations where nucleons transition from lower to higher energy levels. These excitations can occur due to various processes such as nuclear reactions, radioactive decay, or interactions with external particles. There was a lot more available data on these energy levels. I wanted to see if I could predict stability based on the energy levels dataset.

6.5.2 Data

Energy levels data came from the Chart of Nuclides (ChartofNuclides) was used.

The energy density data was calculated from the energy levels data. I counted how many energy levels were present within a given energy range. I created a table containing columns for each energy range, populated with number of levels present. This data was used to plot the number of levels for each N of a particular element, Z. I used this simple approach since it required no knowledge of any existing physics models.
The sample of data below shows 5 energy ranges (I used 1000KeV ranges) and how many energy levels were found in that range. The category was given stable = 1, 0 and 2= not stable.

Z	N	cat_numbers	0	1	2	3	4	5
20	19	2	0	2	8	7	14	16
20	20	1	0	0	3	1	6	14
20	21	2	1	7	20	25	44	50
20	22	1	2	2	10	22	44	42
20	23	1	5	20	33	39	23	12

Figure 31: Sample of energy level data

6.5.2.1 Initial data analysis

The energy densities were plotted for specific Z to understand how the data looked. It can be seen that the levels are densely packed in around specific N values.

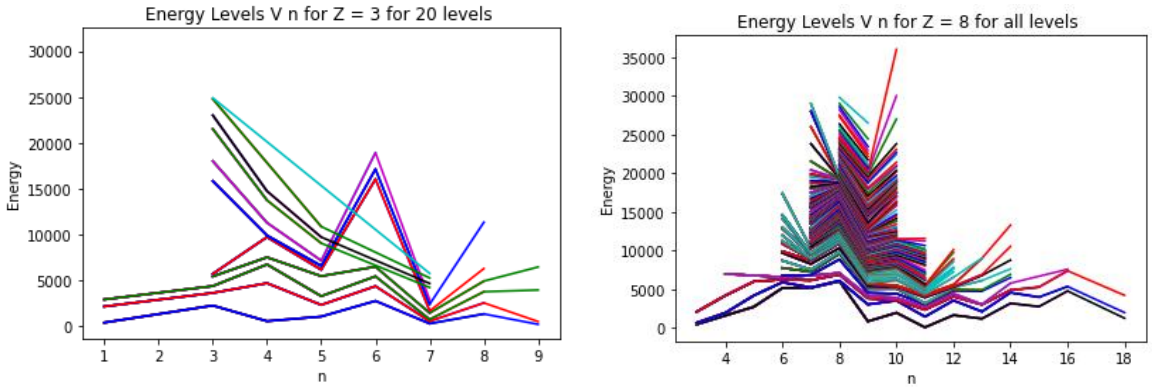


Figure 32: Energy level density increase around magic numbers

The levels per N were counted and plotted. A pattern emerged where the density peaked around the magic numbers.

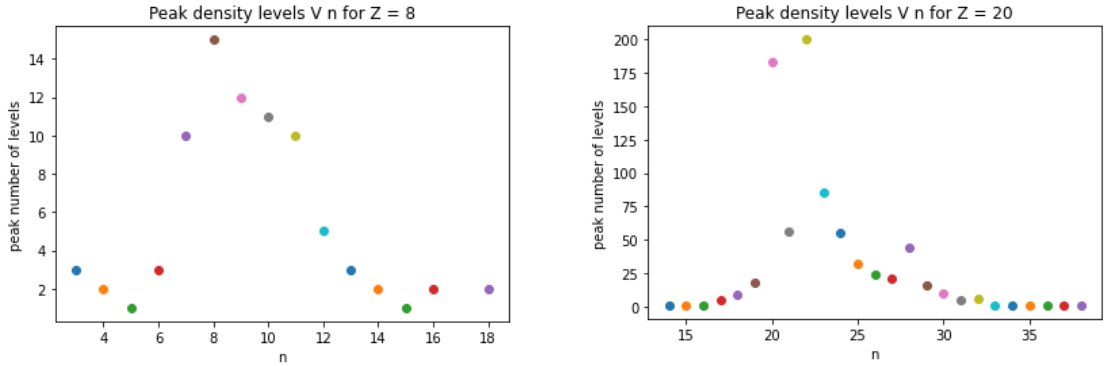


Figure 33: Density peaks around the magic numbers

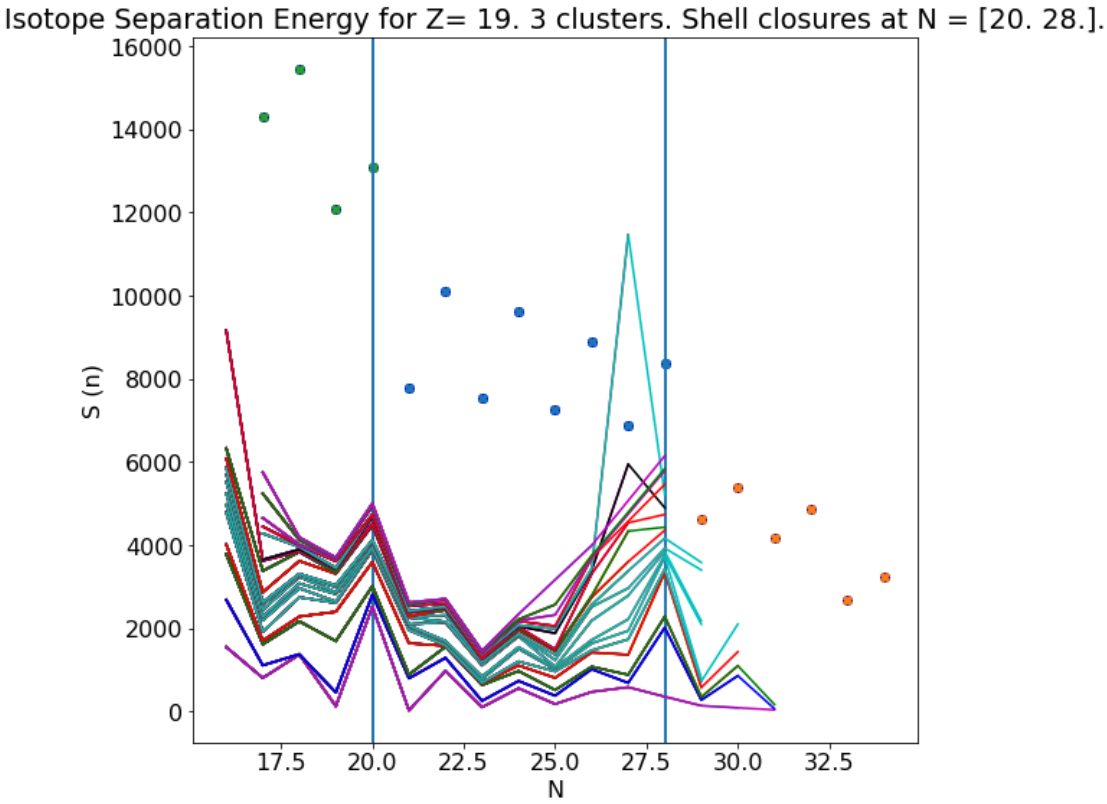


Figure 34: Correlation between energy level peaks and magic numbers

I was able to overlay my energy density diagram on my separation energy diagram to see a distinct correlation between stability and energy density, but could machine learning models identify this same pattern?

For each value of Z, I conducted an analysis to identify the specific N where the density of energy levels peaked. By determining the frequency with which each N value appeared as the location of maximum energy levels, I constructed a histogram to visualise the distribution. This visualisation underscores a consistent pattern in the data, indicating a correlation between stability, magic numbers, and the density of energy levels within a nucleus.
By plotting the frequency of N with highest energy levels, the magic numbers were easily found, indicating that densely spaced energy levels are a clear sign of stability.

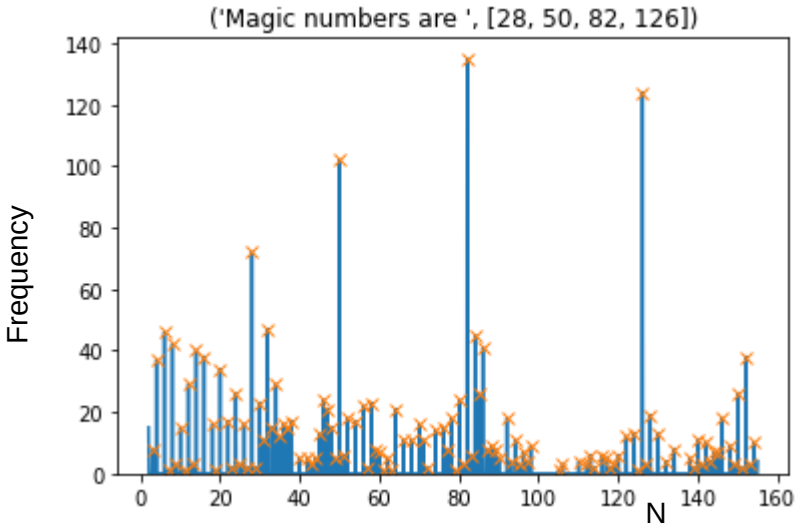


Figure 35:Histogram to show most energy levels for magic numbers

I decided the goal of this experiment was to see if stability could be predicted from energy densities and to investigate the relationship between stability and energy density / energy levels by asking, can we predict stability from energy densities?

6.5.3 Method

I wanted to classify a particular nucleus as either stable or not stable, so this was a binary classification problem. I tried several models and chose the random forest classifier model as being the best. I was using accuracy score to compare.

Once I decided on a model, I next optimised the algorithm by trying out various numbers of 'features' in my data. The data set had over 100 features. Many of the features contained the value '0' so I experimented with leaving some out. I was able to tweak the model until the accuracy score 0.963.

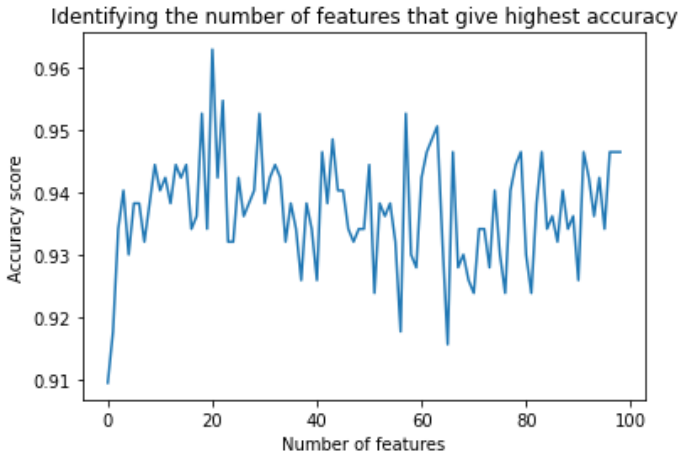


Figure 36: Finding the optimal number of features from a very large data set

Likewise with other tuneable parameters such as ‘estimators’ and ‘rand states’.

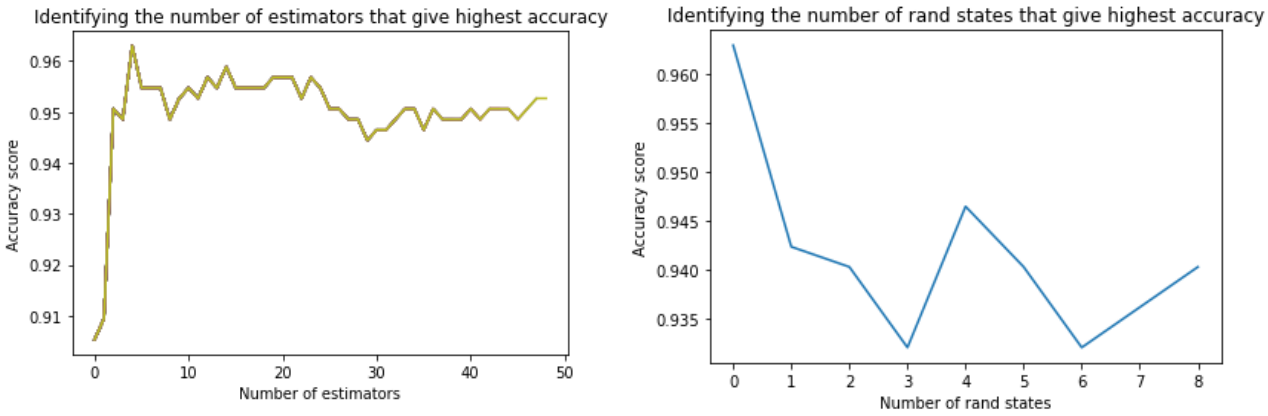


Figure 37: Finding optimal values for tuneable parameters

6.5.4 Results

The random forest algorithm gave the following results with an accuracy of 0.963.

The results were better if I predicted unstable nuclei rather than stable. The accuracy for stable was 0.2857. This was because the data was heavily biased towards unstable data so it was easier to predict if something was unstable. The majority of the data we were using to train the model was unstable. To remove some of this bias a technique called stratification was used but after several experiements it was clear that this technique wasn’t improving the accuracy of predicting ‘stable’ nuclei. I think this was for the same reason as discovered in experiment 4.

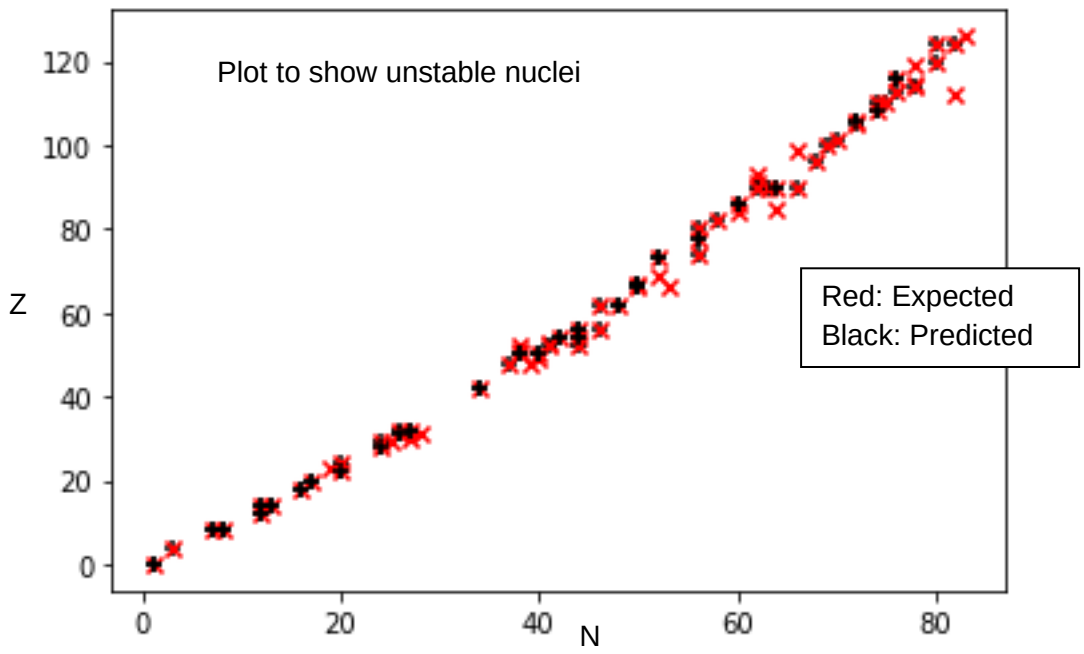


Figure 38: Prediction of unstable nuclei

Feature addition:

It became evident that to be able to predict stability, more features of different types were needed.

I experimented with the addition of spin and magnetic dipole, whilst also experimenting with using just Z or just N.

Z	**N**	**Spin**	**Magnetic Dipole**	**Accuracy**
Y	Y	N	N	28.57
Y	Y	N	Y	25.7
Y	Y	Y	N	20
Y	Y	Y	Y	25.71
N	Y	N	N	31.43
N	Y	N	Y	17.14
N	Y	Y	N	25.71
N	Y	Y	Y	28.57

Table 10: Accuracy of Stability Predictions

The results were inconclusive. I was expecting higher accuracy for the case where I included Z, N, Spin and Magnetic dipole but as you can see no sensible conclusion can be reached with the results.

6.5.5 Conclusion

The data isn’t complete, we may only have density data for nuclei that have been measured so a lack of density might not mean the data doesn't exist, it could mean it just hasn't been measured so we need to remember this when carrying out analysis.

Trying to predict the classification of something in the minority dataset needs more input. There just isn’t enough data for the model to predict a stable nucleus with any reasonable degree of accuracy.

This highlights the need for datasets that are more balanced and contain a good spread of data for all the classifications you may want to predict.

The interesting point to note here is that we can easily see the magic numbers with a bit of data analysis, machine learning was not required.

6.5.6 Further work

Another interesting point to note was during further analysis I saw some heavier elements being predicted as stable. This may have been due to their half life being really large, e.g. for Uranium it is 4.5 billion years. I would have liked to study this area further to see how often heavier elements were predicted to be stable and to understand what pattern that would show up as.

6.6 Experiment 6: Predicting Spin

6.6.1 Overview

This experiment was to see if machine learning could identify the pattern between spin and Z and N and potentially even-even and stability.

6.6.2 Data

Using the same set of data from Chart of Nuclides (ChartofNuclides) I used Z, N and spin. Python needed numbers as float or integers, so I turned the spin values to floats to give the following data.

Spin as float - all data total 1545 data points.

0.0 872
1.5 120
2.5 119
0.5 104
3.5 85
1.0 60
4.5 58
2.0 45
3.0 25
5.0 18
4.0 15
6.0 9
7.0 8
5.5 5
8.0 2

These numbers were then turned into categories in preparation for the classification model (it works by assigning categories). There were 15 categories.

One problem became clear immediately. The data is very heavily biased around spin = 0 data. To level out the categories I reduced the number of categories to 3. *Spin as 0, odd or even.*

Spin class	Count
0	872
2	602
1	1

We can see class 0 and class 2 are a bit closer in size but class 1 is still really small in comparison.
Using feature engineering I created the target column of ‘Type’ which was either a 1 for boson nucleus (integer spin), 2 for fermion nucleus (half integer spin) or 0 for zero spin, stable nucleus.

6.6.3 Method

This was a binary classification problem, so a random forest classification model was used as this has yielded very good results in the past. However, on reflection I should have continued my method of testing several models at the same since the performance of a models depends on how well it can detect underlying patterns and that really depends on the data it is being trained on. Just because a model may have performed well on one data set it doesn’t mean it can perform well on any data set.

6.6.4 Results

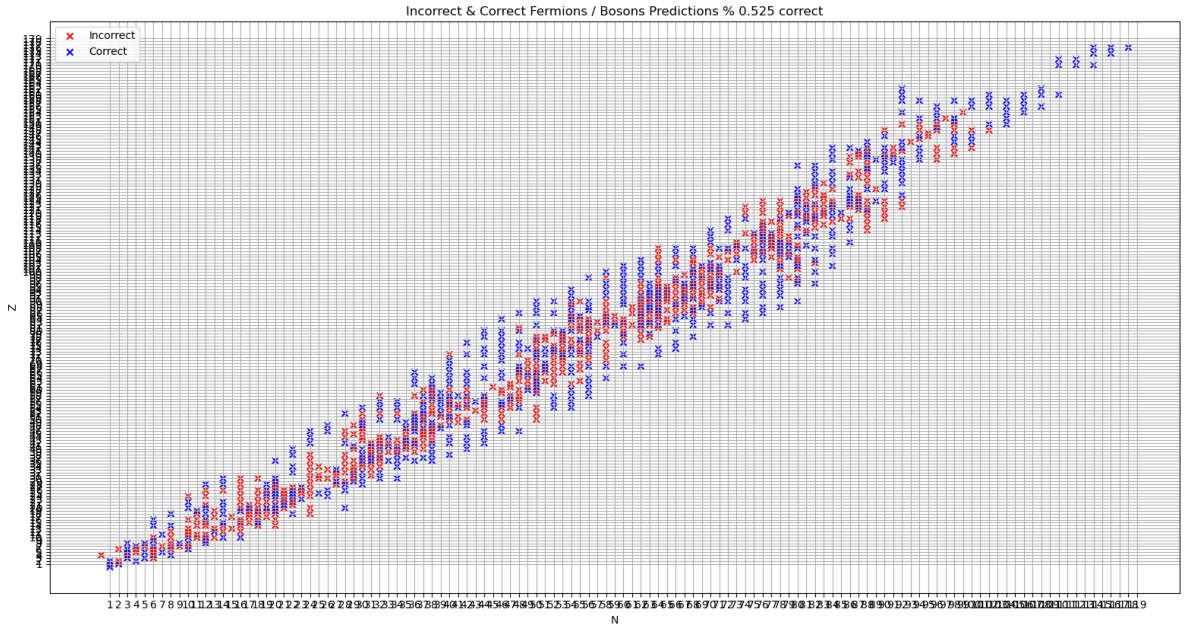


Figure 39:Fermion / boson nuclei predictions

The number of even-even nuclei that were predicted correctly was 55.858%
Out of the 52% correct predictions 76% were correctly predicted as bosons.
I investigated whether this was due to the fact the data might be skewed because it consisted of a higher proportion of bosons.

Data	B	F	Ratio
All data	1002	466	2.15:1
Test data	752	349	2.15:1
Training data	250	117	2.15:1
Results	645	456	1.41:1

Table 11: Ratio of boson / fermion nucleus in the data set

The ratio was different for the results data, but further investigation is needed to eliminate data skew or understand how it affects data with so few features.
There is more than double the number of bosons to fermions. This means we have an 'imbalanced class dataset'. There is an uneven distribution of the data.

A high percentage of the predictions were correct, but we must look more closely at the type of errors that were made during the classification process.

To do this we create a confusion matrix.

First, we must calculate the number of correct predictions for each class.

Correct fermions 309 (True positive)

Correct bosons 651

Then we calculate the incorrect number of predictions for each class, organised by predicted value.

Incorrect bosons as fermions 101 (False positive).

Incorrect fermions as bosons 40.

The values are then arranged in a confusion matrix.

	B	F
B	**651 (TP)**	40 (FP)
F	101 (FP)	**309 (TP)**
Total	752	349

Table 12: Confusion matrix

More errors were made when predicting bosons as fermions than when predicting fermions as bosons.

The model favours the majority class.

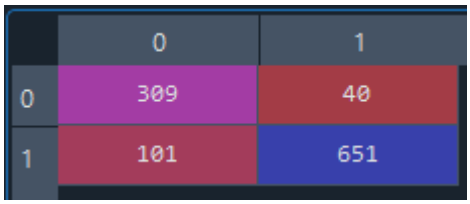


Figure 40: Confusion matrix showing real data

With the past experiences of unbalanced data, I tried again to improve the results. With the decision tree we can use balanced bagging classifier.

	0	1
0	319	28
1	97	657

Figure 41: Decision tree bagging 88.6% accurate

	0	1
0	302	45
1	96	658

Figure 42: Random forest, no bagging, 87.1% accurate

There was not much difference between the two models, but the errors are more balanced in the random forest bagging. 97 / 28 errors vs 96 / 45.

I also used SMOTE as a way to oversample, but this didn't improve the results.
When the results don't improve it's a strong indicator that the model needs more data to help it identify the underlying patterns.
I added in another feature of magnetic dipole.

z	n	magnetic_dipole	Spin as f/b
92	138	0	1
80	130	0	1
60	87	0.554	0
104	156	0	1
16	20	0	1
37	46	1.4249	0
10	24	0	1
92	136	0	1
38	60	0	1
63	96	1.38	0
56	83	-0.973	0
54	84	0	1
42	58	0	1
17	18	0.8218743	0

Figure 43: Data with magnetic dipole.

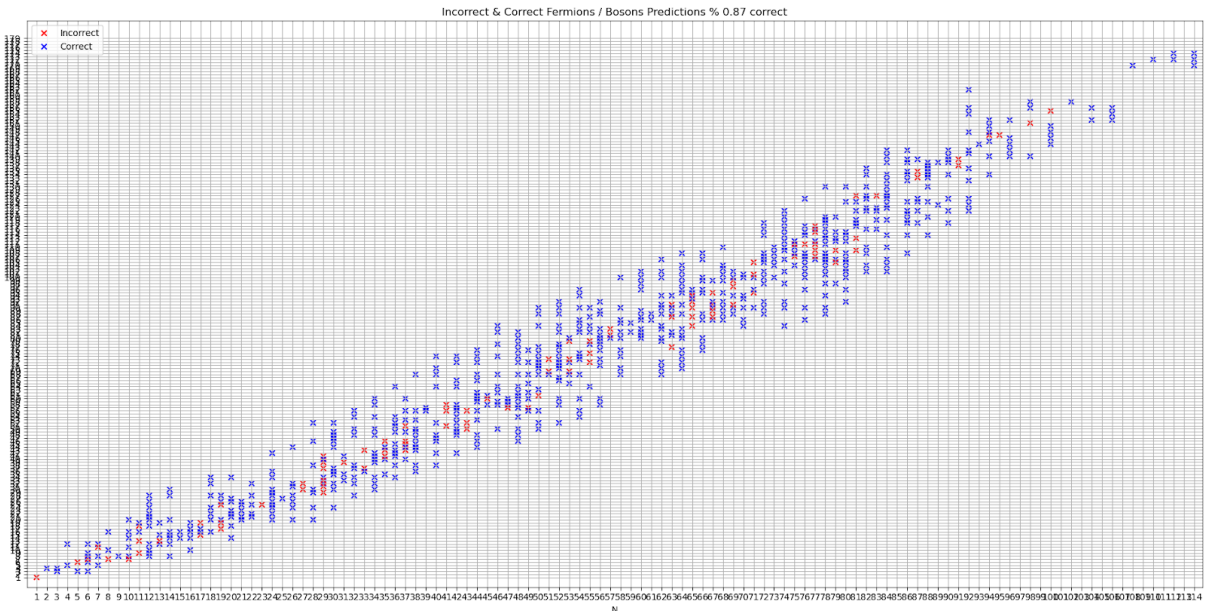


Figure 44: Slightly improved fermion / boson nuclei predictions

The number of even-even nuclei that were predicted correctly was 61.875%

6.6.5 Conclusion

Even-even nuclei are more stable and therefore more easier to observe so the data on them is more widespread. This also makes them well-studied, so the models are well developed for predicting deformity in these nuclei. The huge amount of data on them made them easier to predict compared to none even-even nuclei.

Odd -odd or odd-even are more difficult to study so theoretical data is somewhat limited resulting in them being harder to predict..

6.7 Experiment 7: Identifying the Magic Numbers from Deformity

6.7.1 Introduction

The majority of nuclei undergo some degree of deformity. A non-deformed nucleus has a spin of zero. This experiment was to see if machine learning could identify the pattern between spin and stability. Can we identify the magic numbers in deformity data?

This was a data analysis activity rather than a machine learning activity. I included it to show that we can still learn a lot from data without the need to apply complicated machine learning models.

6.7.2 Data

I used Z, N and the deformity parameter Beta from NuDat (NuDat).

n	z	quadrupoleDeformatio
4	6	0.824521
6	6	0.576601
14	6	0.403011
8	6	0.359384
12	6	0.300075
10	8	0.365498
8	8	0.353042
12	8	0.268063

Figure 45: Raw data of Z, N and deformation parameter Beta

For each Z, I took the N with the lowest deformation value. I then counted up all the Ns and plotted a histogram.

```
(20.0, 0.1821056317329294)
(20.0, 0.1566899813346762)
(20.0, 0.1624594303360171)
```

Figure 46: Data showing N = 20 appearing three times with low values of deformation

6.7.3 Results

The magic numbers 20, 28, 50 and 82 can be seen from the histogram proving the relationship between stability and low values of deformation.

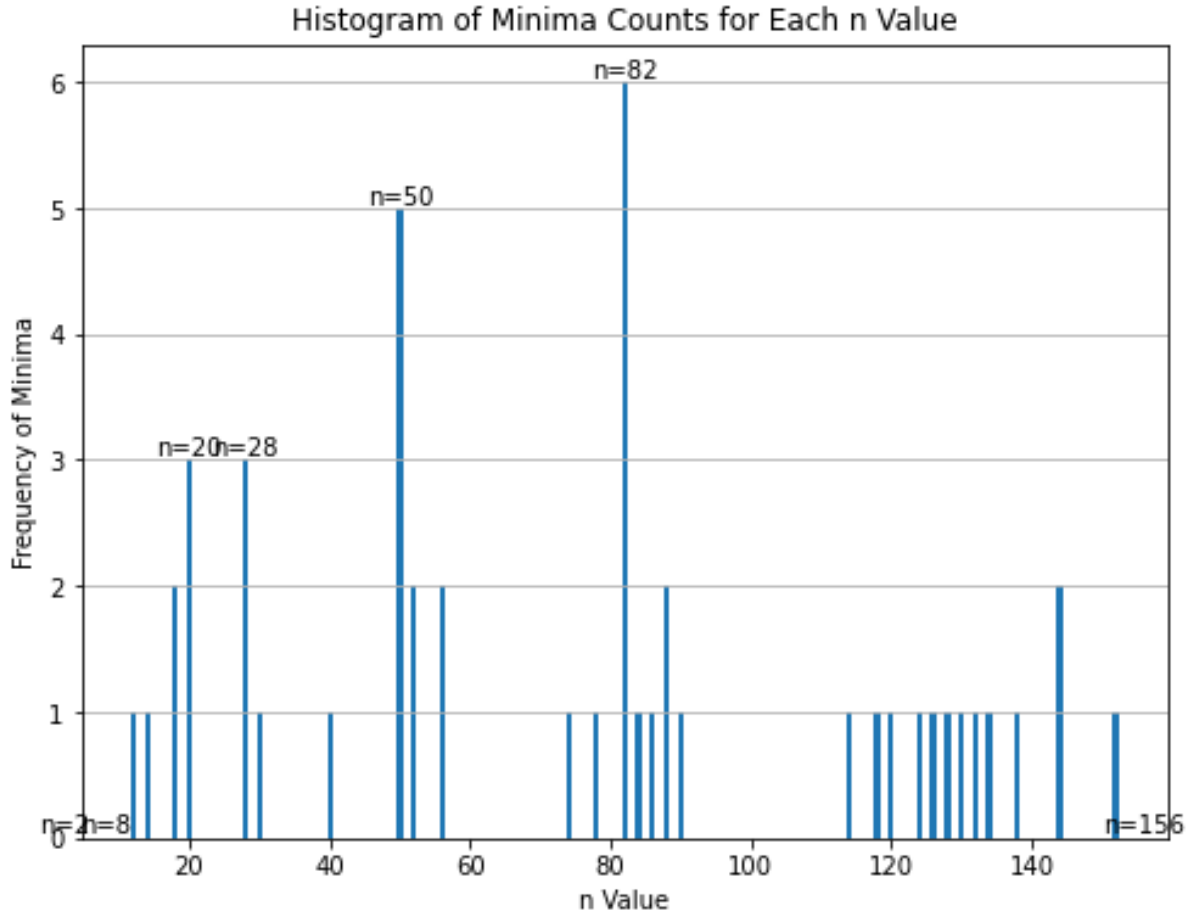


Figure 47: Histogram proving the relationship between deformation and stability

6.7.4 Conclusion

This exercise was a quick data analysis activity to show that some patterns in data can be easily found without machine learning. I would like to emphasise that the easiest solution will always be best, and we shouldn't over complicate data analysis just because we can.

6.8 Experiment 8: Predicting Deformity from a Theoretical Geometric Model

6.8.1 Overview

After learning about the shell model and deformation I formulated an idea based on the nucleus being built gradually in clusters of 2 protons and 2 neutrons as a Helion or alpha particle. When 4 nucleons come together tightly packed, they are shaped as a tetrahedra.

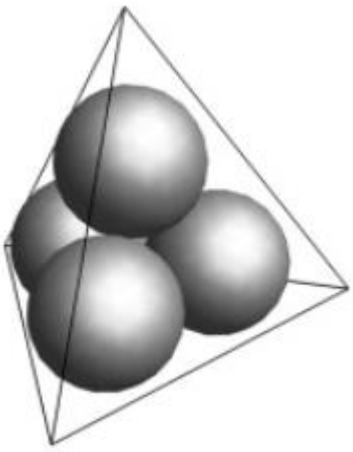


Figure 48: Possible formation of an alpha particle

To try and explain deformity I imagined energy as waveforms forming nodes where they met with waveforms from another nucleon. My idea is that these nodes may cause the appearance of deformity since the energy waves will be concentrated around the nodes.
The more alpha particles that were joined together the more nodes that were created and the more pronounced the deformity. This can also be seen as the heavier a nucleus is the more deformed it may be.

In the geometric image below, I have added 3 alpha particles together and put red spots where I the nodes are. I have drawn an oval through these nodes to suggest the shape of the nucleus.

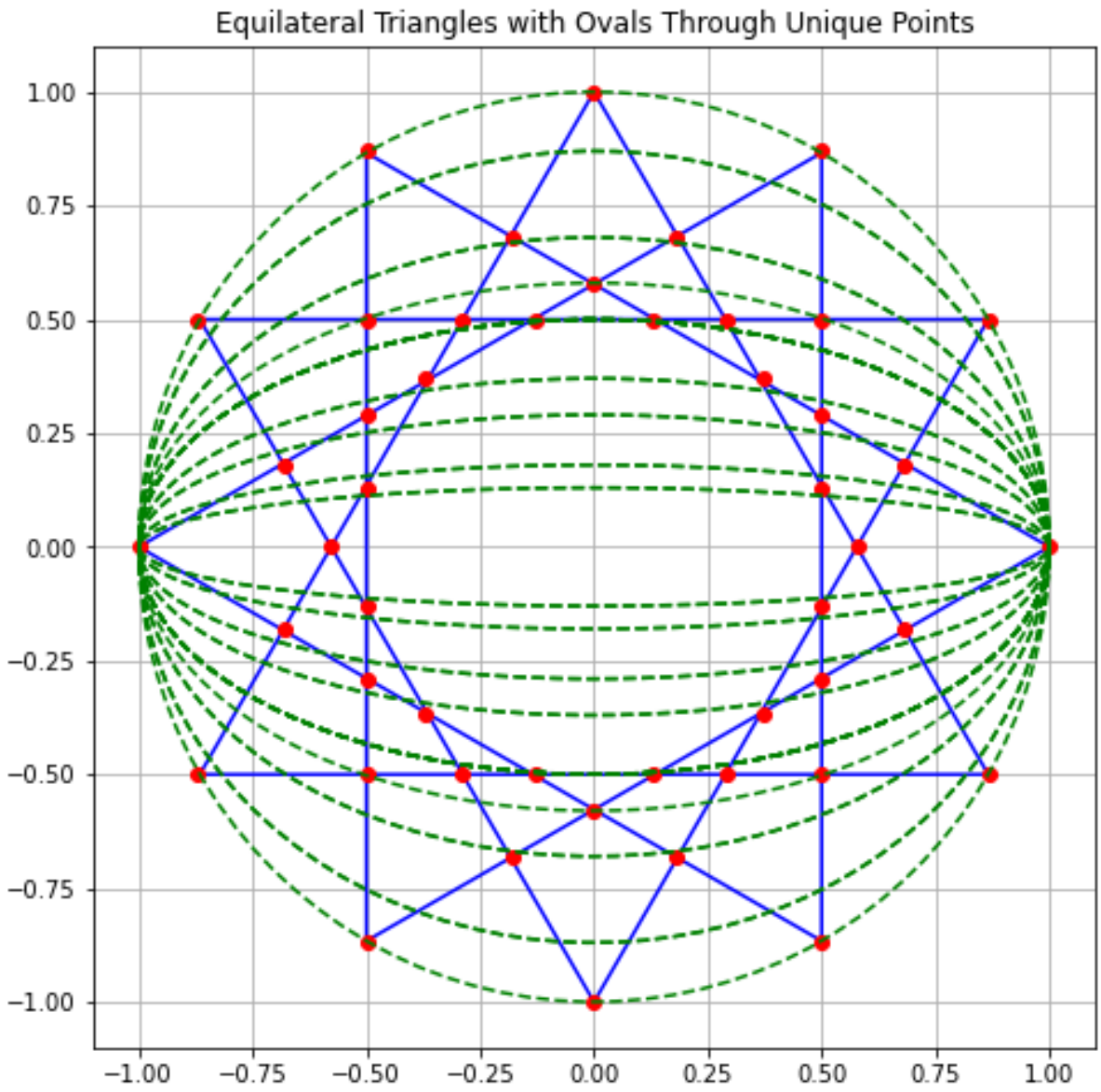


Figure 49: Three alpha particles and their nodes suggesting a deformed shape

I know the physics isn't exactly clear in this model, but the goal of machine learning is to see if we can identify a pattern, or relationship without knowing too much about the underlying physics.

6.8.2 Data

Step 1: Building a model for how many tetrahedrons exists in each shell or layer.

I built up my alpha particles in shells similar to the shell model. This didn't influence my data, but it did help me to visualise the data set I wanted to create. I created a table with only even neutrons and protons and indicated what shell it would be in, or 'layer' and indicated how many

full alpha particles were present in that layer and how many altogether were present in all the layers.

Shell	P	N	Total	Full helions / tetrahedrons	Accumulative helions / tetrahedrons
1s	2	2	4	1	1
1p1	4	4	8	2	
1p2	2	2	4	1	4
1d1	6	6	12	3	
1d2	4	4	8	2	9
1f1	8	8	16	4	
1f2	6	6	12	3	16
1g1	10	10	20	5	
1g2	8	8	16	4	24

Table 13: Layered helion model data

Step 2: Building a theoretical dataset of the geometry of deformed circles based on the number of triangles and therefore nodes. Also, I used the fact that deformation was related to eccentricity.

6.8.3 Method

With my data set I performed clustering to give me possible distinct deformed shapes.
I took the average distance to each cluster and calculate the eccentricity.
I then tried to use this to predict deformation. I then compared that value to real nuclei deformation values.

Deformity is a continuous value, so the problem is a regression problem. I used gradient boost and random forest regression models to try and identify patterns between my data set and deformity.

6.8.4 Results

The random forest regression results were not very good.

Root Mean Squared Error: 0.12351384540630445

R-squared: 0.043946831880670056

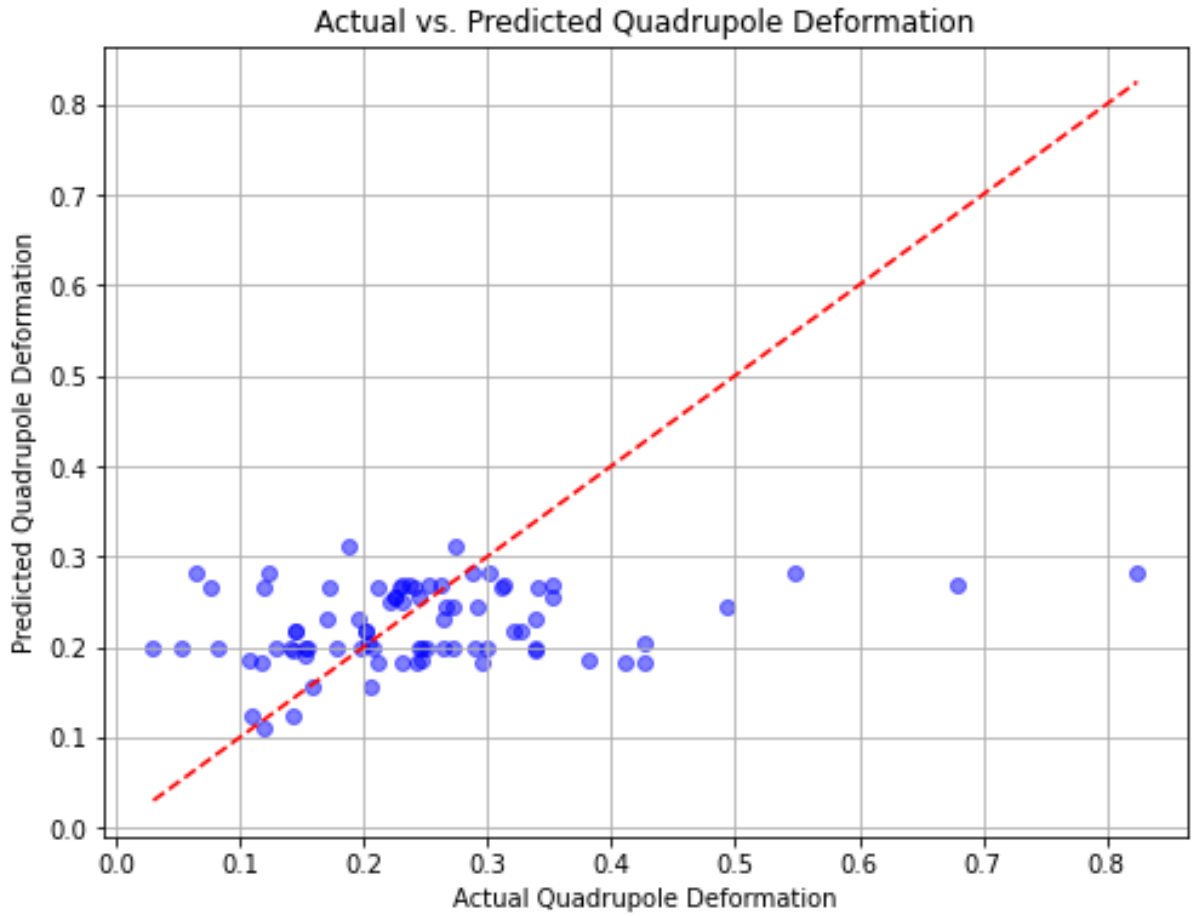


Figure 50: Random forest regression results

The results from the gradient boost model were equally as poor.

Root Mean Squared Error: 0.12537071764318913

R-squared: 0.014984685620576643

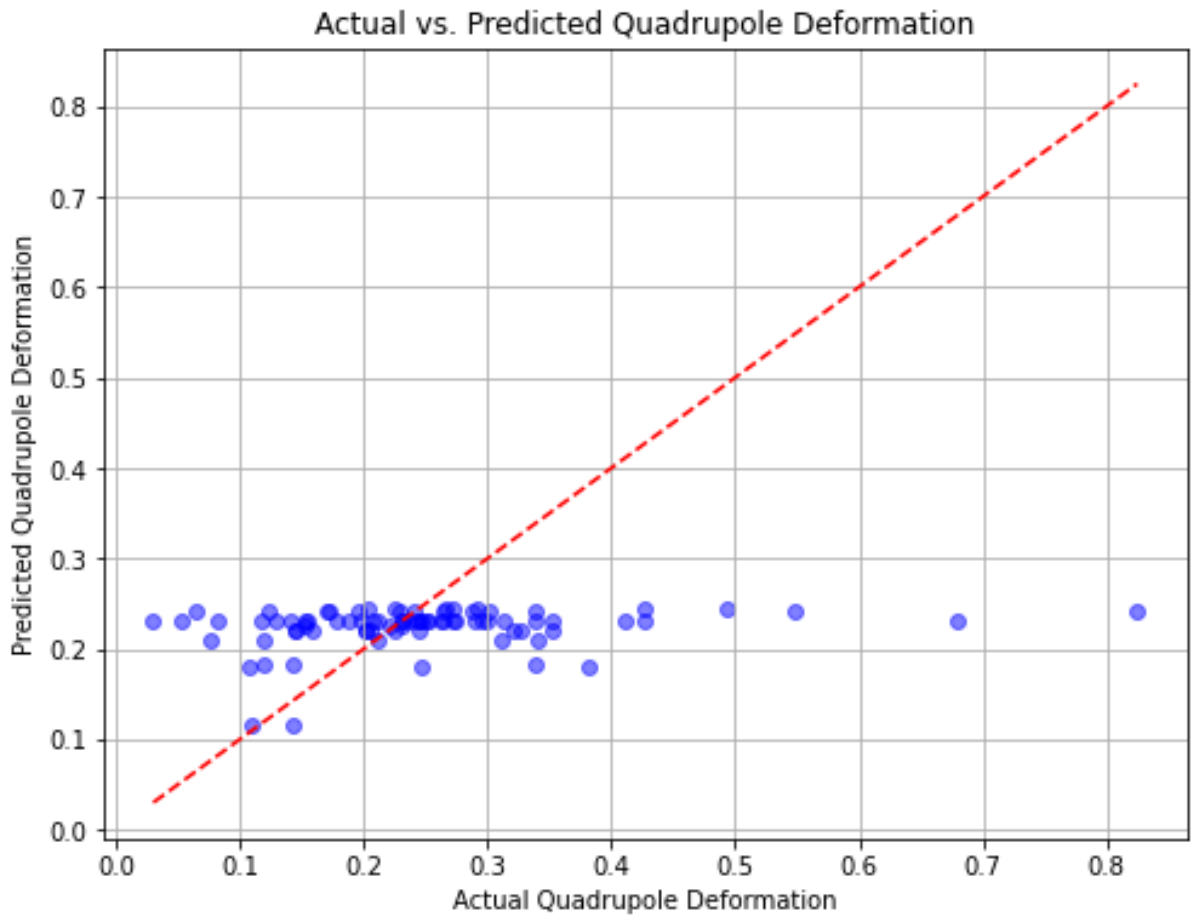


Figure 51: Gradient boost results

Neither model was able to identify a relationship between the node shapes and the deformation parameter.

6.8.5 Conclusion

This was my attempt to create a novel model of the nucleus, using geometry as key feature. There are several areas where my data could be improved. I have used a 2D representation for my nodes. Perhaps, this is too much of a simplification. If I could translate this to a 2D model, it would be more representative. As always, accurate data is paramount, and I just don't think my data was good enough.

It is clear there are other forces in play in the nucleus that cause the deformed shape, so it is hard to be completely naïve about these. I think ignoring current models as I have been doing is the wrong approach.

For machine learning to be fully embraced we need to see it as a tool to support our current thinking and to help us find new questions. The answering of those questions can only come with specialist knowledge, supported by machine learning, not replaced by it.

Machine learning should help us uncover questions and challenge our thinking. It can help us on the road to discovery.
I think we do need a lot of knowledge to understand the data, so it is still something to be used by people who are specialists in the subject matter.

7 Chapter 4: Final Conclusion and Further Work

Throughout the experiments in this thesis there was a common theme around the lack of quality data, and this affected the quality of the results.

Experiment 2 demonstrated that predicting separation energies with only three features was not very accurate. This highlighted the complexity of the relationships between nuclear parameters, showing that more features are required to capture these interactions effectively. The need for additional data underscores the challenge of identifying and testing the necessary relationships. The quality of the results heavily depends on the availability of the right data. Without knowing which specific data points are crucial, achieving the desired outcomes becomes difficult, if not impossible. A broad and diverse dataset is essential for successful experimentation; without it, the experiments are likely to fail. This experiment also showed errors at higher values of Sn which was highly likely due to the lack of data at higher energies due to its instability, making it hard to gather data.

Experiment 3 showed us how small differences in data can make or break an experiment. The S(n) data was slightly more stepped than the S(p) meaning the patterns in the S(n) could be identified much more easily that in the S(p) data.

Experiment 4 showed how the clustering models accuracy suffered due to the small sizes of the clustering and also how it was hard to do regression on small clusters of data.

Experiment 5 showed how data biases lead to the prominent class of unstable being predicted with a high degree of confidence compared to the less prominent class of unstable. A problem when there are not enough classes.

Experiment 6 highlighted the fact that the dataset wasn't complete due to even-even nuclei having longer half-lives, they are more abundant and so can be studied easier and more frequently. Incomplete datasets can give misleading results.

Overall, the limitations I encountered were not related to the machine learning models themselves, but rather to the available data, which was too small, incomplete, biased, or imbalanced. More work is needed to improve the quality of the data by expanding the dataset, ensuring it is more representative, reducing bias, and addressing any imbalances. Enhancing

data collection, preprocessing methods, and curating a more comprehensive and balanced dataset will ultimately lead to better model performance and more reliable predictions.
The experiments in this thesis are using a dataset that is too broad. The experiments should be split into high energy / low energy experiments and high mass / low mass experiments due to the difference in physics that is occurring and the difference in available data at various energy and mass values. If the experiments are focused on a specific zone of the chart of nuclides, instead of using all the available data, we find more meaningful relationships.

8 Bibliography

https://www.nndc.bnl.gov/nudat3. NuDat. [Online]
Jenkins, David G. 2021. Gross Properties of Nuclei. 2021.
Liquid Drop Model. Zelevinsky, Vladimir. 2017. 2017, Physics of Atomic Nuclei, pp. 91-111.
Mayer-Jensen shell model and magic numbers. Velusamy, Ramiah. 2007. 12-24, s.l. : Resonance, 2007, Vol. 12.
Prediction of the shapes of deformed nuclei by the polyspheron theory. Pauling, Linus. 1982. s.l. : Proc. Natid Acad. Sci. USA, 1982, Vol. 79.
Theoretical description of nuclear masses. Litvinov1, Yuri A. 2021. s.l. : EDP Sciences, 2021.
www-nds.iaea.org. Chart of Nuclides. [Online]

9 Literature review

Gross properties of nuclei

In the study of nuclear physics, understanding the gross properties of atomic nuclei is paramount. These gross properties encompass a wide range of characteristics, from nuclear size and shape to nuclear stability and binding energies. Over the decades, researchers have developed various theoretical models and experimental techniques to investigate these fundamental aspects of nuclear structure. Gross Properties of Nuclei (Jenkins, 2021)

The liquid drop model

The liquid drop model, proposed by George Gamow in 1930, treats the nucleus as a drop of incompressible fluid, with nucleons held together by strong nuclear forces. This model successfully explains many macroscopic properties of nuclei, such as nuclear binding energy, nuclear stability, and nuclear fission. For a more in-depth discussion of the liquid drop model and its applications, readers are encouraged to refer to The Liquid Drop Model (Liquid Drop Model, 2017)

The nuclear shell model

The shell model, introduced by Maria Goeppert Mayer and J. Hans D. Jensen in the 1940s, provides a more detailed description of nuclear structure by incorporating the concept of nuclear shells, analogous to electron shells in atoms. According to this model, nucleons occupy discrete energy levels within nuclear shells, leading to the emergence of magic numbers and shell closures. The shell model successfully explains the observed patterns of nuclear stability, nuclear magic numbers, and nuclear excitation spectra. It has become a cornerstone of nuclear structure theory and has been validated by numerous experimental observations. (Mayer-Jensen shell model and magic numbers, 2007)