A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Thais Cristina Glauzer da Silva |
| *Student Number* | sba24101 |
| *Module Title* | Programming AI Concepts |
| *Assessment Title* | CA1 |
| *Assessment Due Date* | 15th December 2024 |
| *Date of Submission* | 14th December 2024 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Abstract

This document is part of the Programming AI Concepts of the Higher Diploma in AI Applications (HCI) at CCT Dublin. The goal is to explore the Glass Class dataset to showcase our learnings in the field of AI applications, namely Python programming for this second CA. This report contains screenshots from the Jupyter notebook used and reflections on each question and their importance for the AI field.

Link to Github containing the Jupyter notebook: <https://github.com/tg9292/CA1-Programming-for-AI>

Contents

[Introduction 1](#_Toc185074850)

[Problem Description 1](#_Toc185074851)

[Objectives 1](#_Toc185074852)

[Methodology 1](#_Toc185074853)

[Neural Network Architecture 1](#_Toc185074854)

[Model Training 2](#_Toc185074855)

[Model Configuration 2](#_Toc185074856)

[Results and Discussion 2](#_Toc185074857)

[Conclusion 7](#_Toc185074858)

[References 8](#_Toc185074859)

# Introduction

Accurate classification of glass types is crucial for recycling and quality control processes. This assignment explores the application of neural networks to classify different types of glass based on their chemical composition. Our goal is to showcase the learnings acquired during the Programming for AI module at CCT.

# Problem Description

The primary challenge lies in accurately classifying glass types based on features like refractive index and chemical components. Traditional classification methods may struggle with the complex relationships between these features and the glass type. Neural networks, with their ability to learn complex patterns, offer a promising solution.

# Objectives

The primary objective of this study is to develop and implement a neural network model capable of accurately classifying glass types based on the given features. Specifically, the model aims to:

1. **Data Preparation:** Clean and preprocess the data to ensure it's suitable for neural network training.
2. **Model Architecture:** Design and implement a neural network architecture that effectively learns the underlying patterns in the data.
3. **Model Training:** Train the neural network on the prepared dataset, optimizing its parameters to minimize classification errors.
4. **Model Evaluation:** Evaluate the trained model's performance on a separate test dataset to assess its accuracy and generalization ability.

# Methodology

**Data Preparation:**

1. **Data Loading:** Load the glass data from the CSV file into a Pandas DataFrame.
2. **Data Exploration:** Analyse the data to understand its distribution, missing values, and outliers.
3. **Data Cleaning:** Handle missing values (e.g., imputation) and outliers (e.g., capping or removal) if existent.
4. **Feature Scaling:** Standardize the features to a common scale to improve model performance.
5. **Data Splitting:** Divide the dataset into training and testing sets to train and evaluate the model.

# Neural Network Architecture

A neural network with multiple hidden layers is employed for this task (Chowdhury, 2023). The architecture consists of:

1. **Input Layer:** Receives the input features (refractive index, sodium, magnesium, etc.).
2. **Hidden Layers:** Multiple hidden layers with ReLU (Rectified Linear Unit) activation functions extract complex features from the input data.
3. **Output Layer:** A softmax layer outputs probabilities for each glass type.

In the specific case of glass type classification, a neural network can effectively learn the subtle differences in chemical composition that distinguish one glass type from another. By training the network on a sufficiently large and diverse dataset, we can achieve high accuracy and reliability in classification.

# Model Training

1. **Model Compilation:** The model is compiled with the Adam optimizer and sparse categorical crossentropy loss function.
2. **Model Training:** The model is trained on the training data using a specified number of epochs and batch size.
3. **Model Evaluation:** The trained model is evaluated on the testing data to assess its performance.

# Model Configuration

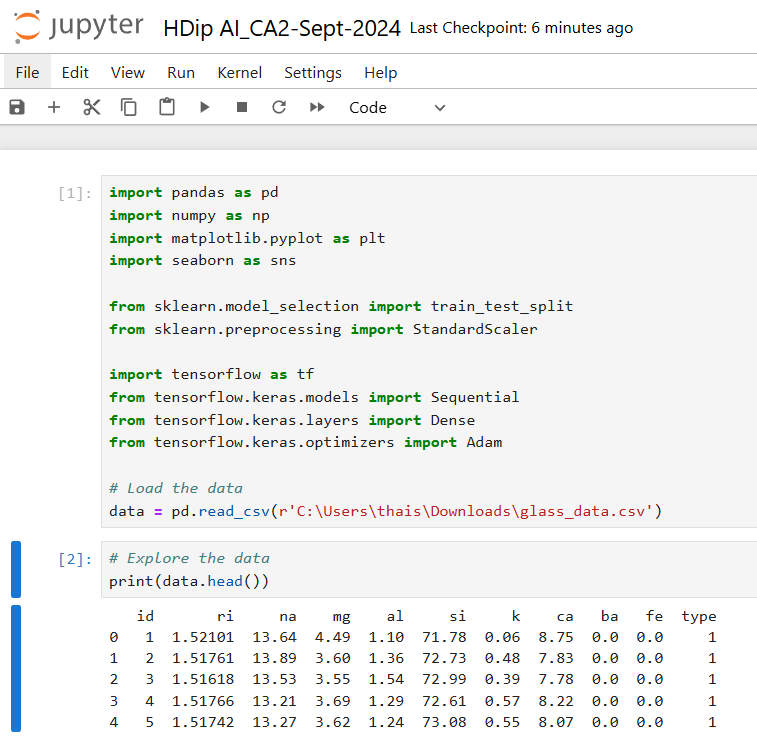
* **Layers:** The number of hidden layers and neurons per layer can be adjusted to optimize performance.
* **Activation Functions:** ReLU is used in the hidden layers to introduce non-linearity, while softmax is used in the output layer to produce probability distributions over the glass types.
* **Loss Function:** Sparse categorical crossentropy is used to measure the difference between the predicted and true class labels.
* **Optimizer:** The Adam optimizer is used to update the model's weights during training.

# Results and Discussion

To better illustrate the results, each step of the Python code used will be provided along with a brief explanation of the choices made during the study and its conclusions.

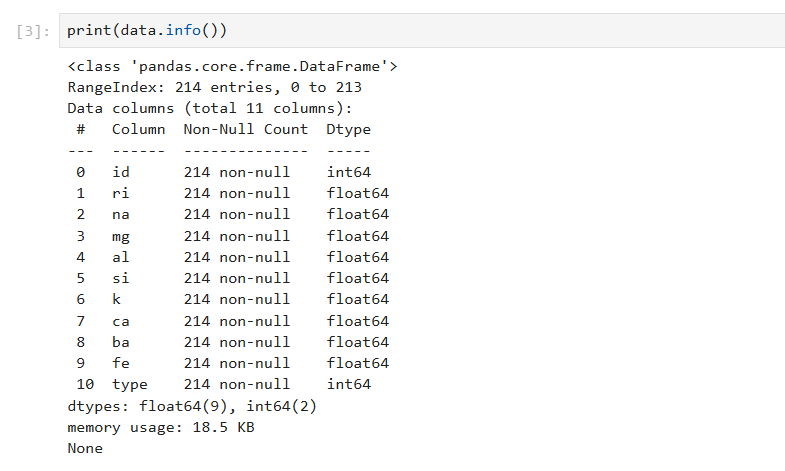
[1] We’ve loaded all libraries needed to support our model. These are required to shorten our code as they contain pre-written formulas to help a shorter and more intuitive use of Python. Along with the libraries, we’ve included the path to the dataset used.

[2] When understanding your dataset, there are critical steps to be followed to ensure you cover all data preparation required. We’ve started here by displaying the first 5 rows of our dataset (i.e. head). It’s a quick way to visualize the amount of columns you have along with the data formats and a sample of potential challenges (such as missing values).

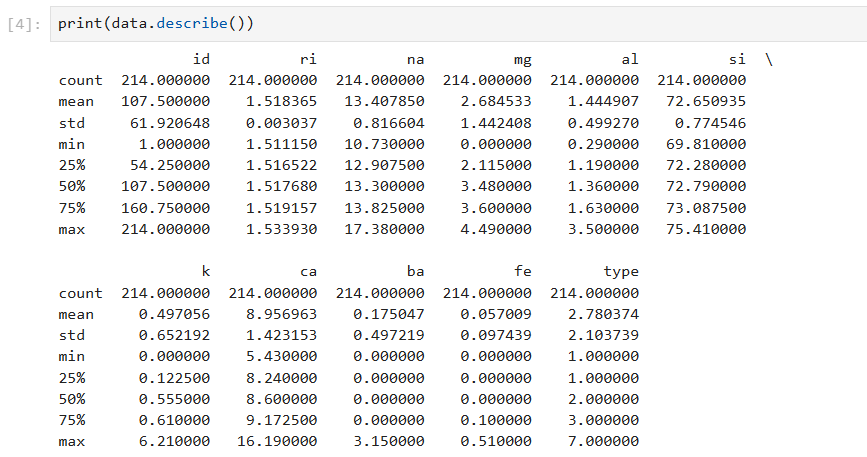


[3] Data info shows the format of the data. In the field of machine learning it’s fundamental since models cannot understand strings. In our example all fields are numerical, making it easier to process as there’s no need to convert strings to numbers.

Another important piece of information is the null count. We can see that all rows in our data are provided. There’s no need to further clean any rows or columns with missing values for this study.



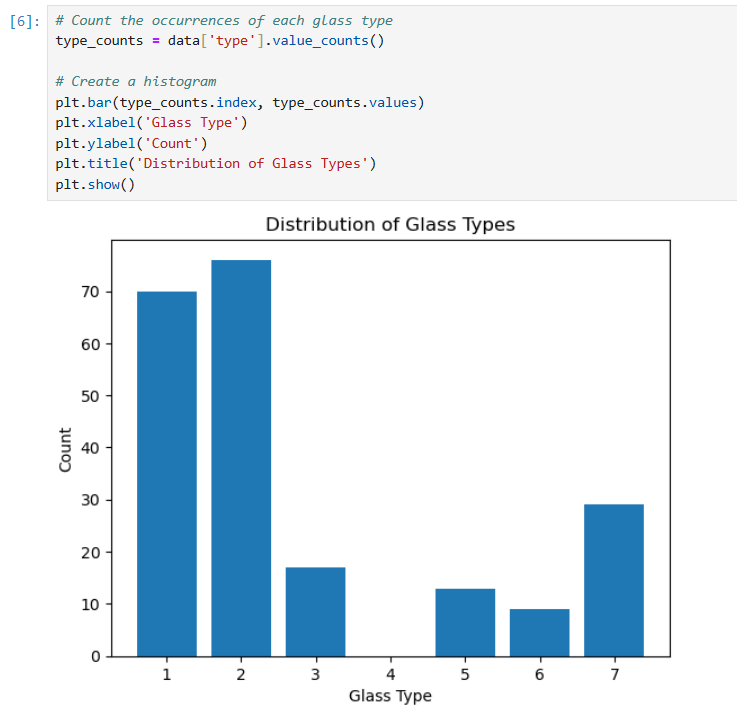
[4] A data description helps us understand the discrepancies in our values by displaying count, mean, max and min we can have a good view of the ranges where our data lies. This is an important step when looking for anomalies and outliers in you sampled population.



[5] Counting each type of id in your dataset helps to spot imbalances in your populations. This is relevant because an imbalanced dataset can lead to biased results in machine learning. Luckily there are techniques to tackle negative outcomes associated with imbalanced datasets (such as SMOTE).

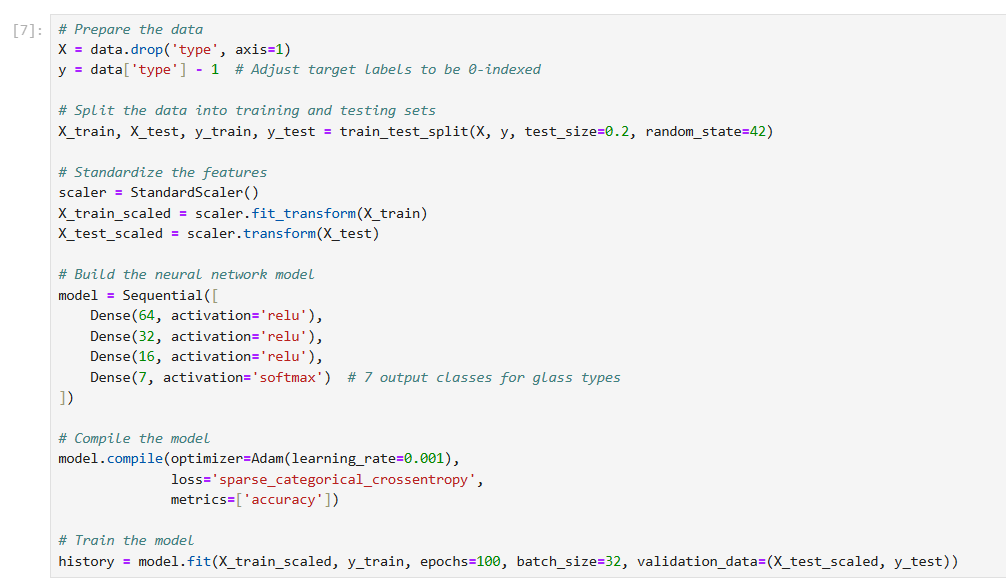


[6] We’ve decided to show the same information provided in step 5 in a chart to make visualization easier for the users of this study.

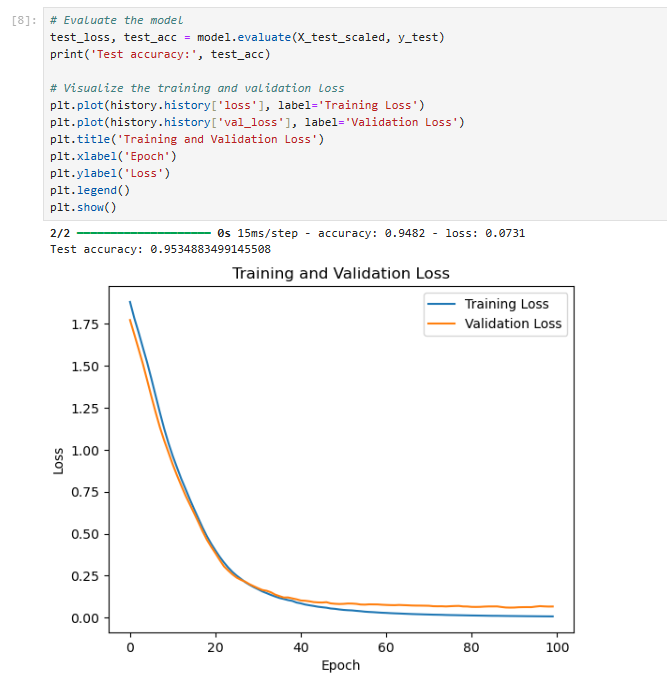


[7] We removed the heading of our table to run the model, this is needed because the heading is not part of the data per se and is in string format. Then we proceeded to split the dataset for model training, we are using 20% for testing size. Fitting the scaler calculates the mean and standard deviation of each feature. This ensures consistency in the feature space between training and testing sets.

We built the model using a specific number of neurons (e.g., 64, 32, 16) in each layer, applied an optimizer rate of 0.001 (which is the most common standard) and trained the model



[8] Once trained, we evaluated the model performance and displayed the visual for training and validation loss. With a resulting accuracy of 95.34%, we can say our model ran successfully. Further metrics can also be explored depending on the needs of your study (such as precision, recall and f1 scores). The training and validation loss help us understand if our model is overfitting, underfitting or just right for our data. In our case the model is slightly overfitting because the training loss decreases steadily, indicating that the model is learning the training data well and the validation loss also decreases initially, but it starts to diverge from the training loss after a certain number of epochs. This suggests that the model is becoming too complex and is starting to memorize the training data rather than learning generalizable patterns (Géron, 2019).



# Conclusion

This report demonstrates the effectiveness of neural networks in classifying glass types based on their chemical composition. The developed model achieved 95.34% accuracy on the test set and is slightly overfitting. Future work may involve exploring more advanced neural network architectures, such as convolutional neural networks or recurrent neural networks (Idrees, 2024), to further improve performance.

# References

Artasanchez, A. and Joshi, P. (2020) *Artificial Intelligence with Python*. 2nd edn. Mumbai:

Packt Publishing. ISBN: 9781839219535

Chowdhury, P (2023). *Neural Networks: Basics of Architecture*. Available at: <https://medium.com/@praneshchowdhury/neural-networks-understanding-the-basics-of-architecture-31d2cf2afd1a>. Accessed on: 14th December 2024.

Géron, A. (2019) *Hands-On Machine Learning with sci-kit learn, Keras and TensorFlow:*

*Concepts, Tools and Techniques to Build Intelligence Systems*. 2nd edn: Updated to

TensorFlow2. New York: O'Reilly. ISBN: 9781492032649

Idrees, H (2024). *ANN vs. CNN vs. RNN vs. LSTM: Understanding the Differences in Neural Networks.* Available at: <https://medium.com/@hassaanidrees7/ann-vs-cnn-vs-rnn-vs-lstm-understanding-the-differences-in-neural-networks-94486cbb6d5a>. Accessed on: 14th December 2024.

Johansen, A. (2016). *Python: The Ultimate Beginner’s Guide*. Leipzig, Germany: Amazon Distribution GmbH.

McClure, N. (2017) *Tensor Flow Machine Learning Cookbook*, Birmingham: Packt

Publishing. Database: eBook IT Core (EBSCO)

Raschka, S. and Mirjalili, V. (2019) *Python Machine Learning: Machine Learning and Deep*

*Learning with Python, sci-kit learn and Tensor Flow 2*. 3nd edn, Birmingham: Packt

Publishing. ISBN: 9781789955750

Russano, E and Avelino, E.F. (2019*) Fundamentals of Machine Learning Using Python*. ON:

Arcler Press. Database: eBook IT Core (EBSCO)