CA2 - Programming with AI

| **Module Title:** | Programming with AI |
| --- | --- |
| **Student Full Name:** | Barbara Weltson Leite da Silva |
| **Student Number:** | sba24096 |
| **Assessment Due Date:** | 15th of December |
| **Date of Submission:** | 15th December |
|  |  |
|  |  |

[**Introduction 2**](#_uyye6cd8zsn8)

[Problem description 2](#_585a8kqh9u0n)

[Objective 2](#_kiq6kgrhaj4s)

[EDA 2](#_xh0560ngj9f0)

[Data Preparation 2](#_eeic8kbkz7i4)

[Below are the steps described for data preparation: 2](#_qfqzuhwnnp0m)

[Model configuration and analysis 3](#_ujb1sjz6o13j)

[After testing different layers, rates, batch sizes and training epochs, this was the final model selected: 3](#_g90qhsjk94p3)

[Hyperparameter Optimization 4](#_cfaj7xzbo3fp)

[Hyperparameter Impact Analysis 4](#_g4n66b47rs5v)

[**Classification with Neural Network 5**](#_cmw1xcz4yu8q)

[Model Performance 5](#_fx3ahavy4gsz)

[Conclusion 5](#_80on7uygm2zq)

[**References 6**](#_zfmlokhl528m)

[**AI usage 6**](#_j41rexrssc4w)

# Introduction

Glass is a material that is heavily used in several industries and serving a range of different purposes. But not all glass is created the same way as it depends on its objective . Identifying the type of glasses is important to ensure the quality expected is achieved and to distribute the glasses accordingly to each industry

## Problem description

The task provided is to classify the glass based on chemical features and refractive index, and associate it with a specific category. The target variable type is what represents the type of the glass.

## Objective

The aim of this document is to classify different types of glasses based on their properties while using a neural network model. In this document, I’ll be adding initial discoveries based on a first review of the dataset, including data cleaning, preparation and the reasons why this was selected for the neural network model development.

## EDA

The dataset provided contains 214 rows and 10 columns, with columns:

* na, mg, al, si, k, ca, ba, fe being numerical columns that represent the chemical of the glass (sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron content respectively.)
* Ri being the refractive index
* id being the identifier of each column
* Type being the target variable
* All columns are numerical

From an initial point, we can also see that the distribution of the target variable is imbalanced, with the majority of the types being 1 and 2.

## Data Preparation

## Below are the steps described for data preparation:

The first steps taken for Data preparation were checking if there was no missing data. Since all columns are numerical, no review of categorical data was required.

I’ve also dropped irrelevant columns like id, since this is a unique identifier and does not have relevance to the model. With this we can simplify the dataset and the analysis for the model.

Additionally, we can visualize the target variable distribution. The reason we do that is to check if there’s any class imbalance before developing our model. In this case, we can see a significant imbalance which can affect the neural network model by having imbalance data presented. In this case we might need to introduce oversampling in the model to try to avoid imbalance in data.

I’ve also added boxplots to help analyse the data. Boxplots assist with identifying outliers. Outliers can have an impact in any model.

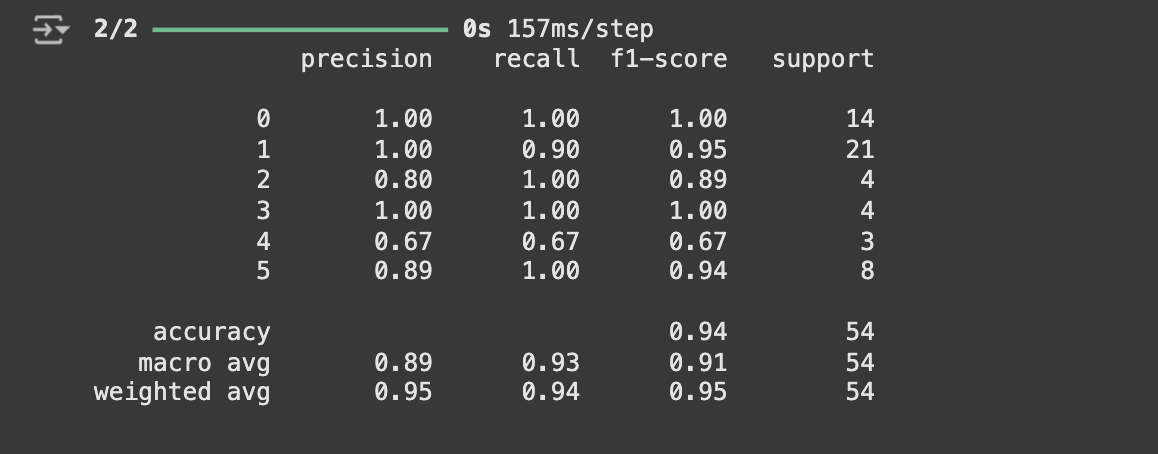
We’ve also introduced feature scaling. If features have very different ranges, the features that present larger values can impact the model as it can make features with lower values harder for the model to detect, and make training for the Neural Network harder. For this, the StandardScaler library was used.

## Model configuration and analysis

## After testing different layers, rates, batch sizes and training epochs, this was the final model selected:

* Input Layer: 32 neurons with ReLU activation
* Hidden Layer: 16 neurons with ReLU activation
* Dropout layers (0.3)
* Output layer: softmax activation
* Learning rate variations: Adam with learning rate=0.003
* Neuron configurations: 32 to 128
* Dropout rates: 0.2
* Batch sizes: 16
* Training epochs: 30

The configuration above was selected after trying with different configurations. I’ve also tried adding more hidden layers (up to 3) and changing the dropout layers, but this model has provided me with the best model overall . Also due to the size of the dataset, I didn’t include too many epochs os batch sizes.

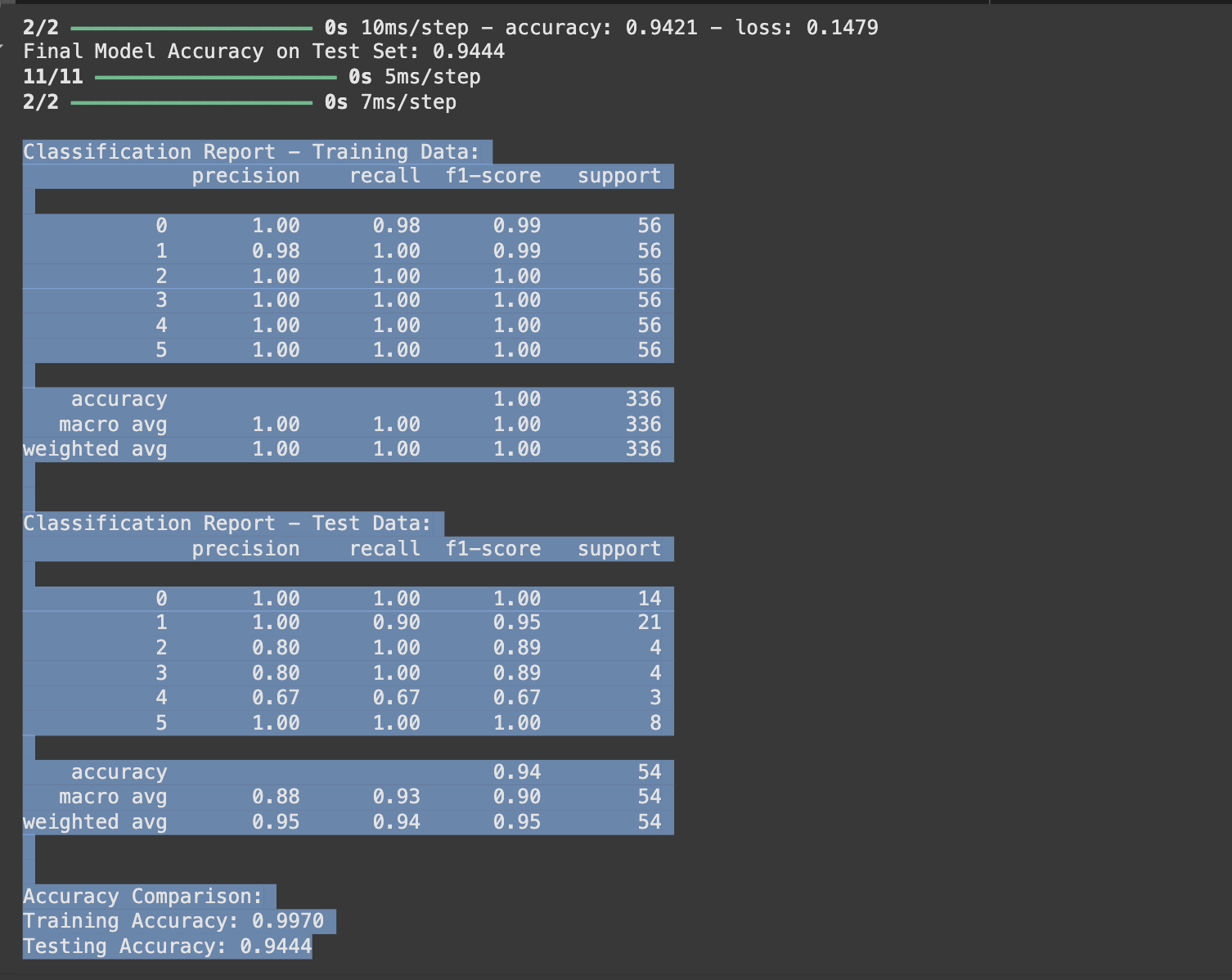


## 

## Hyperparameter Optimization

When running the hyperparameter optimization, the model found was the following:

* Neuron configurations: 128
* Dropout rates: 0.3
* Batch sizes: 32
* Training epochs:50



### 

## Hyperparameter Impact Analysis

* The first difference we can observe is the difference in the network size, with a difference from 32 to 128 neuros on the hyperparameter option. This provides a higher capacity for learning but there’s a risk of overfitting due to having too many neurons
* The dropout rate of 0.3 was kept as since it’s a small dataset
* Epochs were increased from 30 to 50 . This provides more time to epochs to improve as epochs might still be learning around epoch 30.
* Batch size was modified from 16 to 32. I’d still believe 16 could be better in this case due to the class imbalance and the model can check the minority classes, but I also understand SMOTE was utilized to address class imbalance. I also had to modify the k-neighbour from 2 to 3 as not doing so was causing the model to have 100% of accuracy, which was likely being caused by overfitting of the model.

# Classification with Neural Network

A simple NN configuration was applied to the glass type dataset. The model was trained on both the training dataset and the test dataset and evaluated on both.

## Model Performance

* On the training set we had an accuracy of 99.70%
* On the testing set we had an accuracy of 94.44%

The difference of 5.26% indicates that there is a good generalization ability of the model and that there’s no overfitting in the model.

**Challenges:**

The dataset contained class imbalance, which could have impacted model performance. In order to address this, SMOTE was applied to oversample minority classes, in order to improve classification performance.

## **Conclusion**

In this project, I created a Dense Neural Network to classify glass types using different features based on the provided dataset. This involved preparing the data, testing different model setups such as testing with different number of epochs, bataches and dropout rates and , and improving the performance by tuning the hyperparameters and using techniques to avoid overfitting and to address class imbalance.

The final model worked well and with a higher accuracy than I expected, as shown by the accuracy on both the training and testing data. While the training accuracy was a bit higher, the difference wasn’t big. This shows that the model didn’t overfit too much. At first, the model was giving me 100% accuracy which is extremely rare, so I had to go through the model to ensure that the parameters were correct. I modified the input layer from 64 to 32 to help with this step.

One important step was tuning the model’s hyperparameters, like the number of neurons, learning rate, and batch size. This made the model perform better and faster. I also used SMOTE to balance the dataset because some classes had fewer samples. This helped the model learn to classify all types of glass more fairly and not just focus on the most common ones.

In summary, the model did a good job classifying glass types, and I learned a lot during the process. A lot of the concepts provided in class I can say that only started making more sense after putting this into practice, and by implementing a Neural network, I now have a deeper understanding of the overall concepts.

## References

Chia, A. (2024). *Python Boxplots: A Comprehensive Guide for Beginners*. [online] Datacamp.com. Available at: https://www.datacamp.com/tutorial/python-boxplots.

Usama (2022). *Classification Report*. [online] kaggle.com. Available at: https://www.kaggle.com/code/usamabajwa86/classification-report.

Vignesh, S. (2020). *The Perfect Fit for a DNN.* [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/the-perfect-fit-for-a-dnn-596954c9ea39.

## AI usage

* Gemini was utilized in collab to help with errors (such as import errors)
* ChatGPT was used to check the last report

Github Repo

https://github.com/babiweltson/ca-2-programming-for-ai

Recording

https://capture.dropbox.com/Wv946A8uxU8XxU97