A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name:* | Kevin Scully |
| *Student Number* | 24275 |
| *Module Title* | Programming for AI |
| *Assessment Title* | CA2 |
| *Assessment Due Date* | 15th December 2024 |
| *Date of Submission* | 13th December 2024 |
| *Word Count* | 1,257 words |
|  |  |

**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.

Contents

[Introduction 1](#_Toc184998062)

[Problem Description & Objectives 1](#_Toc184998063)

[Methodology 1](#_Toc184998064)

[A. Data Preparation 1](#_Toc184998065)

[i) Nil (0.00) values 2](#_Toc184998066)

[ii) Drop columns 2](#_Toc184998067)

[iii) Outliers 2](#_Toc184998068)

[iv) Preparing the Data for the Model 4](#_Toc184998069)

[B. Developing the Model 4](#_Toc184998070)

[i) Initial Neural Network 4](#_Toc184998071)

[ii) Improving the Dense Neural Network 4](#_Toc184998072)

[iii) Hyperparameter and Cross Validation 5](#_Toc184998073)

[iv) Architecture Diagram 5](#_Toc184998074)

[Results & Findings 6](#_Toc184998075)

[References 7](#_Toc184998076)

[Appendix: 8](#_Toc184998077)

Introduction

The dataset provided contains the classification of chemicals used to create a type of glass. This project aims to build a dense neural network model to classify the different types of glass based on the make-up of the different chemical properties. The model will be compared at different stages while using different parameters. The model will be developed and hyper-parameterised over a three-week period, with adjustments made up until the project deadline to achieve the best possible accuracy.

Problem Description & Objectives

The ‘glass data’ provides the breakdown of nine different elements that are used to make a type of glass. There are 214 rows and 11 columns on the dataset. The columns are all numerical, with ‘type’ being categorical. The columns are ID, the nine chemical elements (a column for each chemical element) and the type of glass as a category. The classification task is to predict the correct glass type based on the given chemicals. A dense neural network will be used to predict the type of glass by identifying and learning patterns within the chemical composition.

The objective of this project is to develop a dense neural network-based classification model that can accurately predict the type of glass based on the chemicals.

Methodology

## Data Preparation

The initial analysis of the data showed that there were no null values or duplicated rows. There was also no clear correlation identified between any columns, except for when there was a high percentage of nil values.

A screenshot of a data analysis

Description automatically generated

### Nil (0.00) values

The description of the dataset showed that the columns ‘mg’, ‘k’, ‘ba’ and ‘fe’ had a minimum value of 0.00. There were no minus values within the dataset. It is assumed that the 0.00 values were accurate as it is possible that there was 0.00 of the elements in the formation of the glass type. As these were accurate values, it was decided for the model to only handle the values if they were extreme outliers. (Tukey, 1977) The below graph shows the percentage of the 0.00 values in each column:

A graph of values per column

Description automatically generated

### Drop columns

As the column ‘id’ had no impact on the glass type, this was dropped from the dataset. This would ensure that the column did not impact the model. Additionally, due to the high number of zero values and the low variance in the ‘ba’ column, this was also dropped from the dataset. Low variance columns provide little information and could impact the performance of the model. (Mehta, et al., 2019)

### Outliers

As the information within the dataset was accurate in the creation of the glass type, only extreme outliers were identified and handled. (Tukey, 1977) The outliers were identified using the interquartile range (IQR). The upper bound was calculated using plus 3 and lower bound was calculated using minus 3 in order to identify only extreme outliers to the lower and upper bounds. The results are shown in the below graphs:

A group of white squares with blue dots

Description automatically generated

The extreme outliers were handled by being replaced with either the mean or median to bring them within the bounds. To determine whether the mean or median should be used, the skewness of each column was calculated. If the data was moderately skewed to asymmetric (skewness was between -1 and +1) then the mean was used, and if the data was highly skewed then the median was used. (Hubert & Van der Veeken, 2008) The below graphs show the skewness of each column:

A group of graphs showing different types of data

Description automatically generated

The extreme outliers were replaced using the mean / median for each column. When these values were replaced, the upper and lower bounds would change, pushing values outside of bounds and causing them to become extreme outliers. When this happened, the skewness was re-checked to ensure the correct measurement was being used (mean or median) and the extreme outliers were replaced. This was repeated until there were no extreme outliers within the dataset. Removing the extreme outliers allows the dense neural network to perform better by reducing noise and overfitting. (Jabbar & Khan, 2015)

### Preparing the Data for the Model

The columns were split into feature and target variables. The target variable was ‘type of glass’ which was a multiclass classification. The data was then split into training and testing sets, with 30% of the data reserved for testing. As the target variable has a multi-class classification, One-Hot Encoding was used to transform the numeric labels into a binary format that the neural network can process the target variable effectively. The data was then ready to be passed through the machine learning model.

## Developing the Model

## Initial Neural Network

The initial neural network consisted of three dense layers, two of which were hidden layers that had 64 and 32 neurons respectively. The final layer had 6 outputs as there was 6 classes in the target variable. The activation function Relu was used as it has been shown that it converges much more quickly and reliably than other activation functions. (Ahmed & Longo, 2022) L2 regularization was added to each layer to reduce overfitting by penalizing large weights. An additional step used to reduce overfitting was a dropout layer, which randomly sets 20% of the units to zero to reduce the reliance on certain neurons. This model used a SoftMax activation function, which is suitable for multi-class classification as it outputs probabilities for each class. (Martins & Astudillo, 2016) Finally, the model was compiled with a categorical cross-entropy loss function and the Adam optimizer. The Adam optimizer adapts the learning rate during training for a more efficient convergence. (Tong, et al., 2022) The model was fit with 100 Epochs, a batch size of 32 and a validation split of 20%. This produced a poor accuracy of 29%.

## Improving the Dense Neural Network

To ensure that the feature columns were contributing equally to the dense neural network, the features were scaled using standard scaler. This ensured that all features were on the same scale. The target variable was then checked for distribution. The below graph shows that the class is imbalanced. Synthetic Minority Over Sampling Technique (SMOTE) was applied to rectify this class imbalance in the training set.

A graph of different colored squares

Description automatically generated

SMOTE ensures that each class is equally distributed, as shown by the distribution below after SMOTE was applied.

A chart of different colored bars

Description automatically generated

The model was then re-applied, however this time the model was fitted with early stopping with a patience of 10 and Epochs increased to 500. The early stopping prevents overfitting by the training process when the validation loss stops improving. (Hussein & Shareef, 2024) This produced an improved accuracy of 69%.

## Hyperparameter and Cross Validation

GridSearchCV was used to perform a hyperparameter search. This tested the different combinations of activation functions, neuron counts, and optimizers to find the best configuration for the model. This was used with five K-fold validation. This ensured the model was being tuned optimally while also being evaluated in a way that avoids generalisation and overfitting. (Hidayat, et al., 2024) The results showed that the optimal parameters were activation - 'relu', neurons – 256 and optimizer - 'adam'. The test and training data produced similar accuracy, showing that the model generalised well to the new data. The final accuracy was 65% and a best accuracy for a class was 83%.

## Architecture Diagram

A diagram of a computer algorithm

Description automatically generated

Results & Findings

The below graphs compare the performance of the model after each of the steps i), ii), and iii) detailed above across accuracy, precision, recall and f1-score.

A graph of different colored bars

Description automatically generated

The initial model’s low performance of 29% was improved drastically following the balancing of the target variable using SMOTE. The hyperparameter model made slight tweaks to the model, but the performance of the model did not improve.

The precision shows that the model is predicting the correct class 61% of the time. This is higher in the larger classes. shows that the model is performing well when predicting the larger classes. The recall shows that on average the model correctly identified 62% of all true instances. The F1-score shows a balance between precision and re-call, this is being lowered by the smaller classes.

Overall, the model performed adequately with an accuracy of 65%.

# References

Ahmed, T. & Longo, L., 2022. Examining the size of the latent space of convolutional variational autoencoders trained with spectral topographic maps of EEG frequency bands. *IEEE Access,* Volume 10, pp. 107575-107586.

Hidayat, T. et al., 2024. Performance Prediction Using Cross Validation (GridSearchCV) for Stunting Prevalenc. *IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS),* pp. 1-6.

Hubert, M. & Van der Veeken, S., 2008. Outlier detection for skewed data. *Journal of Chemometrics: A Journal of the Chemometrics Society,* 22(3-4), pp. 236-246.

Hussein, B. & Shareef, S., 2024. An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning. *In ITM Web of Conferences,* Volume 64, p. 01003.

Jabbar, H. & Khan, R., 2015. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study).. *Computer Science, Communication and Instrumentation Devices,* 70(10.385), pp. 978-981.

Martins, A. & Astudillo, R., 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. *In International conference on machine learning,* Issue PMLR, pp. 1614-1623.

Mehta, P. et al., 2019. A high-bias, low-variance introduction to machine learning for physicists.. *Physics reports,* Volume 810, pp. 1-124.

Tong, Q., Liang, G. & Bi, J., 2022. Calibrating the adaptive learning rate to improve convergence of ADAM. *Neurocomputing,* Volume 481, pp. 333-356.

Tukey, J., 1977. *Exploratory data analysis.* s.l.:Reading/Addison-Wesley.

***GitHub Link:*** https://github.com/kpscully116/Programming-for-AI-CA2

# Appendix:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

A group of graphs showing different sizes and numbers

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A white background with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A graph of different colored rectangles

Description automatically generated

A screenshot of a computer code

Description automatically generated

A white background with black and white clouds

Description automatically generated