A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Robert Jonjic |
| *Student Number* | sba24456 |
| *Module Title* | Programming for AI |
| *Assessment Title* | HDip AI\_CA2 |
| *Assessment Due Date* | 15/12/2024 |
| *Date of Submission* | 13/12/2024 |

**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.

Contents

[Introduction 2](#_Toc184988030)

[Problem Description 2](#_Toc184988031)

[Objectives 3](#_Toc184988032)

[Methodology 4](#_Toc184988033)

[Exploratory Data Analysis (EDA) 4](#_Toc184988034)

[Data Preparation 5](#_Toc184988035)

[Neural Network Design 5](#_Toc184988036)

[Hyperparameter Tuning 5](#_Toc184988037)

[Evaluation Metrics 5](#_Toc184988038)

[Explanation of Configurations 6](#_Toc184988039)

[Conclusion 6](#_Toc184988040)

[Appendix 8](#_Toc184988041)

**Classification of Glass Types using Neural Networks**

# Introduction

Glass is a highly versatile material used across various industries due to its unique physical and chemical properties. Its applications span from architectural windows and automotive windshields to optical lenses, containers, and tableware. Given the diversity of uses, accurately classifying glass types based on their composition is a critical task in industries like manufacturing, recycling, and quality assurance.

The classification of glass is not only important for ensuring that the correct type of glass is used in specific applications but also plays a vital role in sustainability. For instance, during recycling processes, mixing different types of glass can lead to product defects or inefficiencies in production. Furthermore, precise classification ensures adherence to safety standards, such as in automotive or building windows, where certain types of glass are required to meet stringent regulations.

In this project, we utilize a publicly available dataset that contains nine numerical features related to the refractive index and the oxide composition of glass samples. These features include the proportions of Sodium (Na), Magnesium (Mg), Aluminum (Al), Silicon (Si), and other elements critical to glass composition. The target variable categorizes the glass samples into seven distinct types, such as building windows (float or non-float processed), containers, tableware, and headlamps.

Machine learning methods, especially neural networks, are well-suited for solving classification problems due to their ability to model complex, non-linear relationships in data. Dense Neural Networks (DNNs) are particularly effective for tabular datasets like this one, where the relationships between features and the target variable are intricate.

The primary objective of this project is to develop a DNN model that accurately predicts the type of glass based on its composition. This involves exploratory data analysis (EDA) to uncover patterns in the data, preprocessing to ensure the model is fed with optimized inputs, and hyperparameter tuning to maximize the model's performance. By the end of this project, the goal is to deliver a robust model that can classify glass types with high accuracy while providing insights into the strengths and challenges of using neural networks for such tasks.

# Problem Description

The classification of glass types poses a unique challenge due to the inherent complexity of the material's composition and its applications across diverse industries. Glass is made from a mixture of raw materials such as sand, soda ash, and limestone, combined with specific oxides to achieve desired properties. These compositions determine the glass's refractive index, durability, transparency, and thermal properties, which are critical for its intended use. For instance, the glass used in building windows requires a different manufacturing process and composition than glass used for headlamps or containers.

In the context of the provided dataset, this problem is framed as a multi-class classification task where the objective is to predict one of seven distinct glass types based on its compositional features. These types include categories like float-processed building windows, non-float-processed building windows, containers, tableware, and headlamps. Each class represents a specific industrial application, with precise compositional requirements.

Several challenges make this problem particularly complex:

1. Overlapping Feature Distributions: Some glass types have very similar oxide compositions, leading to significant overlap in feature distributions. This can make it difficult for traditional models to differentiate between classes, particularly for classes like building windows and tableware.
2. Class Imbalance: The dataset exhibits an imbalance in class representation, with certain types of glass (e.g., headlamps) being underrepresented. Such imbalances can lead to biased models that favor the majority classes while underperforming on minority classes.
3. Multicollinearity: Features like Calcium (Ca) and Sodium (Na) exhibit high correlation, which introduces redundancy in the data. Multicollinearity can impact the model's ability to correctly attribute importance to individual features, potentially leading to suboptimal classification performance.
4. Feature Scaling: The numerical features in the dataset, such as oxide proportions and refractive index, vary widely in scale. Models sensitive to feature magnitudes, such as neural networks, require normalization to ensure that no single feature disproportionately influences the predictions.
5. Data Representation: The dataset is composed of purely numerical features, with no categorical or contextual data. While this simplifies preprocessing, it also means the model must rely solely on compositional differences, without access to auxiliary information such as production methods or physical characteristics.

The successful classification of glass types has significant practical implications. For manufacturers, it ensures the correct glass type is used in the right application, reducing waste and increasing product quality. For recyclers, it aids in sorting glass more effectively, promoting sustainability. Addressing the aforementioned challenges requires robust data preprocessing, careful feature engineering, and the design of a neural network capable of capturing non-linear relationships in the data.

By leveraging machine learning and neural networks, this project aims to overcome these challenges and deliver a high-performing classification model, while providing insights into the limitations and potential improvements for future work.

# Objectives

The objectives of this project are as follows, aligned with a structured project timeline to ensure efficient planning and execution:

1. Perform an initial analysis of the data through Exploratory Data Analysis (EDA):
   * Timeline: Day 1-2.
   * Activities: Inspect the dataset for missing values, visualize feature distributions, and generate a correlation matrix to understand relationships among features.
2. Prepare the data for machine learning, including normalization and encoding:
   * Timeline: Day 3.
   * Activities: Normalize the numerical features, encode the target variable, and split the data into training and testing sets.
3. Design and implement a Dense Neural Network for glass type classification:
   * Timeline: Day 4-6.
   * Activities: Develop an initial neural network architecture, test basic configurations, and evaluate preliminary results.
4. Optimize the model's performance through hyperparameter tuning:
   * Timeline: Day 7-8.
   * Activities: Perform systematic hyperparameter tuning using GridSearchCV to identify the optimal number of neurons, activation functions, optimizers, batch sizes, and epochs.
5. Evaluate the model using performance metrics and analyze its predictions:
   * Timeline: Day 9.
   * Activities: Assess the model on test data using metrics like accuracy, precision, recall, and F1-score. Analyze the confusion matrix for misclassification patterns.

# Methodology

## Exploratory Data Analysis (EDA)

The EDA phase involved:

* Data Overview: Checking for missing values, data types, and statistical summaries.
* Visualization: Generating histograms to examine feature distributions and a correlation heatmap to identify relationships between features.
* Findings:
  + No missing values were detected, ensuring data integrity.
  + Features like K and Ba exhibited skewed distributions, which were addressed through normalization.
  + High correlations between features (e.g., Ca and Na) indicated potential multicollinearity.

## Data Preparation

1. Feature Normalization: The StandardScaler was used to scale all features to have zero mean and unit variance, ensuring consistent input ranges.
2. Target Encoding: The categorical target variable (type) was label-encoded to facilitate numerical processing.
3. Data Splitting: The dataset was split into 80% training and 20% testing sets to evaluate model performance on unseen data.

## Neural Network Design

A Dense Neural Network (DNN) was chosen due to its versatility in handling tabular data. The architecture included:

* Input Layer: Equal to the number of features (9).
* Hidden Layers:
  + Two dense layers with 64 and 32 neurons, respectively, activated by ReLU to capture non-linear patterns.
  + Dropout layers (30%) to prevent overfitting.
* Output Layer: A softmax activation function for multi-class classification.

## Hyperparameter Tuning

GridSearchCV was used to optimize the following hyperparameters:

* Number of neurons in hidden layers (16, 32, 64).
* Activation functions (relu, tanh).
* Optimizers (adam, sgd).
* Batch sizes (16, 32).
* Epochs (30, 50).

## Evaluation Metrics

The model was evaluated using:

* Accuracy: Overall correctness of predictions.
* Confusion Matrix: Insights into class-specific performance.
* Classification Report: Precision, recall, and F1-score for each class.

## Explanation of Configurations

1. Neurons and Layers:
   * Hidden layers with 64 and 32 neurons were selected for capturing complex patterns without overfitting.
2. Activation Functions:
   * ReLU was chosen for its simplicity and ability to mitigate the vanishing gradient problem.
3. Loss Function:
   * Sparse categorical cross-entropy, suitable for multi-class classification problems.
4. Optimizer:
   * Adam optimizer provided efficient and adaptive learning rates.

# Conclusion

Interpretation and Discussion of Results

Key Findings

1. Model Accuracy:
   * Training accuracy: 97%
   * Testing accuracy: 94%
2. Confusion Matrix Insights:
   * High accuracy for well-represented classes like 1 (building windows, float-processed).
   * Misclassifications observed for underrepresented classes such as 7 (headlamps).
3. Impact of Hyperparameter Tuning:
   * GridSearchCV improved accuracy by identifying optimal parameters, such as using 64 neurons with ReLU activation.

Challenges

* Class Imbalance: Underrepresented classes affected overall performance, highlighting the need for techniques like oversampling or class-weighted loss functions.
* Multicollinearity: High correlations between certain features may require feature selection or regularization.

Future Improvements

1. Address class imbalance using oversampling techniques or weighted loss functions.
2. Experiment with advanced architectures like Convolutional Neural Networks (CNNs) for feature extraction.
3. Incorporate feature engineering to reduce multicollinearity.

This project successfully classified glass types using a Dense Neural Network. The model achieved high accuracy, demonstrating the effectiveness of hyperparameter tuning and careful data preprocessing. Future work will focus on addressing class imbalance and exploring advanced neural network architectures to further enhance performance.

Word count: 1461

GitHub link: <https://github.com/Robert-Jonjic/PfAI-CA2>

# Appendix

A screenshot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A group of blue graphs

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph with blue squares and numbers

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

Description automatically generated

A screenshot of a computer

Description automatically generated