A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Santhosh Shanmugasundaram |
| *Student Number* | Sba24100 |
| *Module Title* | Programming for AI |
| *Assessment Title* | CA2 |
| *Assessment Due Date* | 15th December 2024 |
| *Date of Submission* | 15th 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.

Contents

[1. Introduction 1](#_Toc185183746)

[2. Exploratory Data Analysis 2](#_Toc185183747)

[2.1 Understand and Adjust the Dataset 2](#_Toc185183748)

[2.2 Visualize and Examine the Relationship Between Variables 5](#_Toc185183749)

[3. Dense Neural Network Model 7](#_Toc185183750)

[4. Hyperparameter Tuning 9](#_Toc185183751)

[4.1 Cross Validation 11](#_Toc185183752)

[5. Classification Report 11](#_Toc185183753)

[6. Conclusion 12](#_Toc185183754)

[7. References 13](#_Toc185183755)

[8. GitHub Link 14](#_Toc185183756)

# Introduction

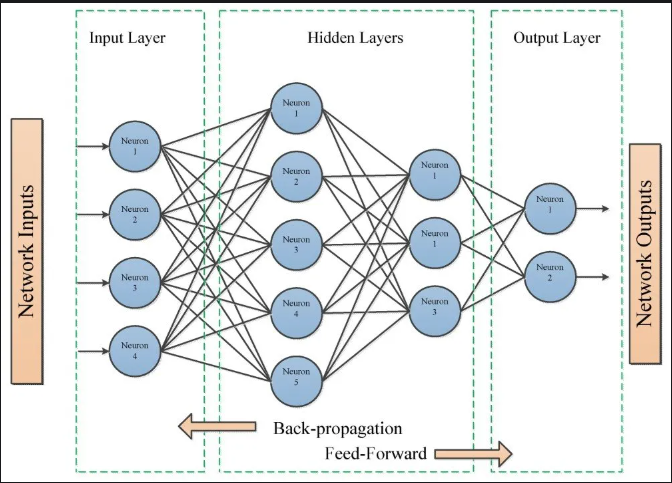
Artificial neural network is a machine learning model which is inspired by the biological neural networks of human brain on how information is processed. In a dense neural network system, there will be one input layer which receives data, one output layer that returns results and there can be 1 or many hidden layers between input and output layer. (Melanie)

Each layer is made up of neurons/nodes and each neuron/node in one layer is connected to the neurons in the previous and/or next layer. Each neurons combine inputs from a dataset with a weight factor and bias factor, to increase or decrease their value. In doing so, neural network system performs calculations.

In addition to neurons, layers, weights and bias, an activation function is performed at each node and a loss function at output nodes. Some of the most used activation functions are RELU, SIGMOID, SOFTMAX and etc. Some of the most used loss functions are ADAM, SGD.

During training of the model, forward propagation and backward propagation activities are performed which modifies the weight and bias factor consistently till model is ready. During testing/prediction phase, based on the weight and bias factors discovered earlier, only forward propagation activity is performed.

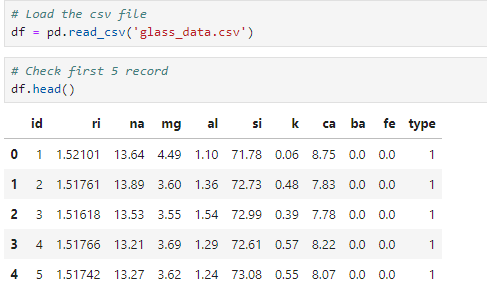
(Corpnce)



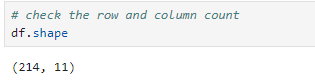
# Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the machine learning process. EDA provides a deep understanding of the data, enabling data scientists to make informed decisions and select the right algorithms for their machine learning projects. It involves examining datasets to uncover patterns, spot anomalies, and test hypotheses before moving on to model building.   
(Tariq A)

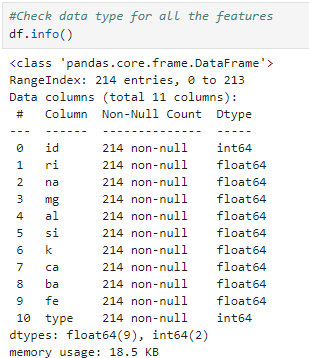
## Understand and Adjust the Dataset

Import the dataset using panda function (READ\_CSV), followed by displaying the top 5 rows of the glass dataset using dataframe function ‘head’.   


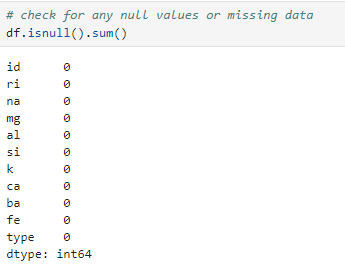
The shape of the dataset can be displayed using function ‘shape’. This helps in identifying the number of rows and columns.



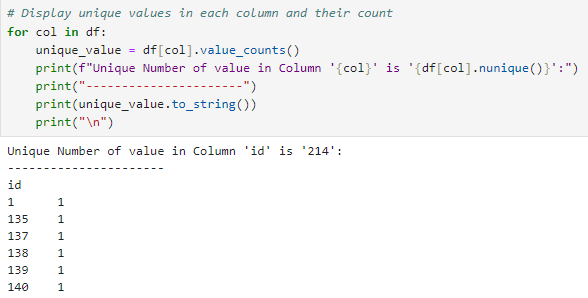
The datatypes of the columns can be checked using ‘info’ function. This tells what type of data is stored in each column.



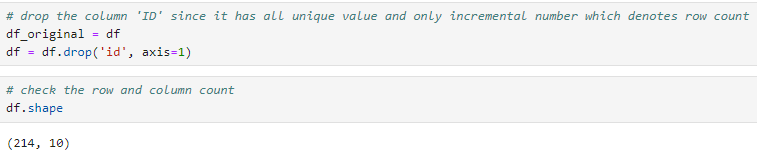
In order to check for any missing values in each column, we can use the function ‘isnull’ and to get the count of missing value, we can use function ‘sum’.



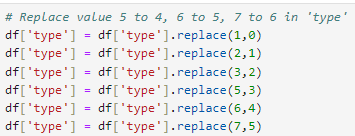
Unique values of specify column can be found using ‘unique’ function of pandas. In a very large dataset, this helps in identifying unique values in each column which is very important to list the possible values.



Checking the unique values in each column, the column ‘id’ contains all unique value in incremental order which acts similar to row number. Column with all unique values is no significance to model training, thereby, this column can be deleted which is used using ‘DROP’ function.



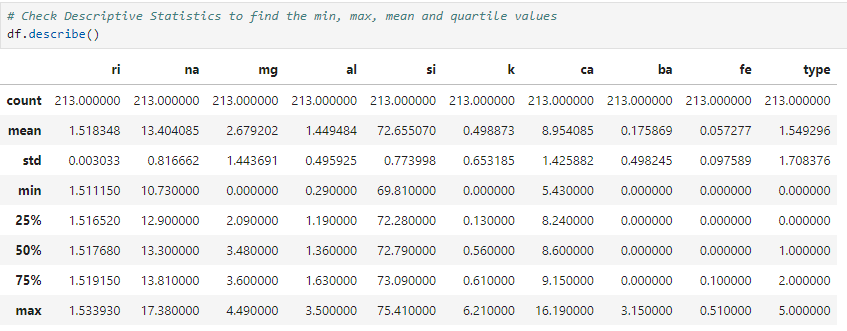
Checking the unique value of the dependent/output column, we noticed that the values start from 1 till 7 with no record for value 4. The loss function ‘SPARSE\_CATEGORICAL\_CROSSENTROPY’ expects the output column to have continuous numbers starting 0. Therefore, the value in the column ‘TYPE’ is replaced to start from 0 with continuous numbers till 5. The same is achieved by using the ‘REPLACE’ function. The result by model prediction needs to be converted back to the original numbers.



It is important to make sure that there are no duplicated rows in the dataset. To verify, we can use function ‘DUPLICATED’ and duplicate rows can be deleted using function ‘DROP\_DUPLICATES’



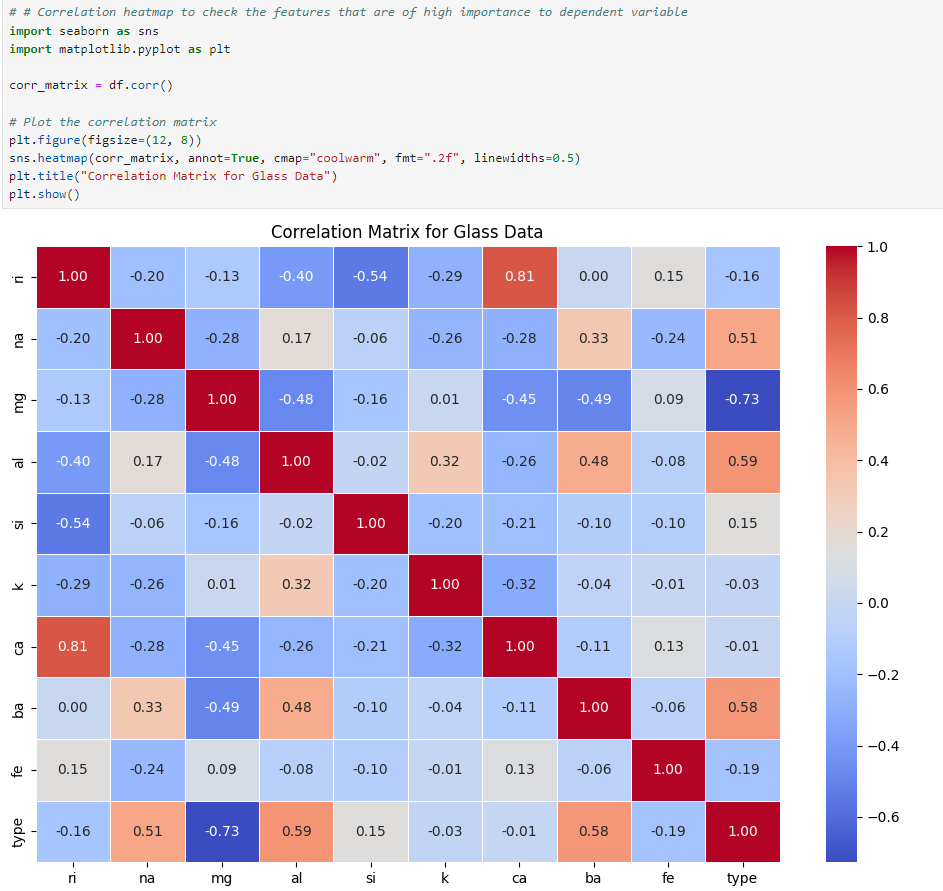
Descriptive statistics of the dataset can be observed using function ‘describe’. This will return the total count, mean, std, min, max, 25%, 50% and 75% percentiles values for each column.   
We can derive the skewness using the difference between the value of two percentiles and outliers can also be derived.



## Visualize and Examine the Relationship Between Variables

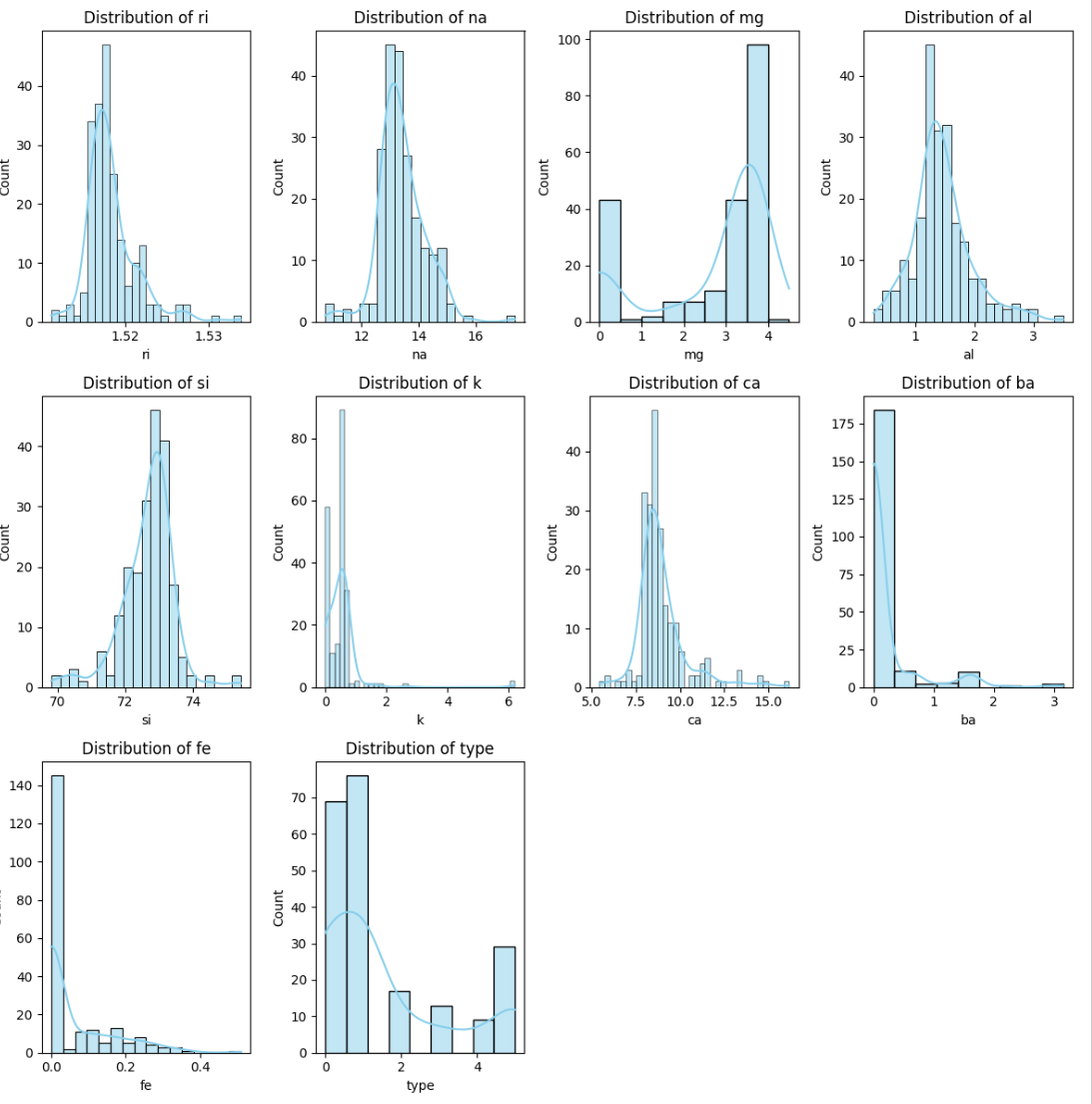
Correlation matrix helps in identifying the relationship between 2 features. Correlation value of 0 indicates no relationship between 2 features. Correlation value of 1 indicates both are strongly proportionate relationship. This means that value for both features increase or decreases at a constant ratio. Correlation value of -1 indicates that both are disproportionate to each other.

Two important libraries namely, SEABORN and MATPLOTLIB is used derive the correlation matrix.



For the given dataset, the dependent variable ‘TYPE’ has positive correlation with variables namely ‘al’, ‘ba’, and ‘na’ and strong negative correlation to ‘mg’.

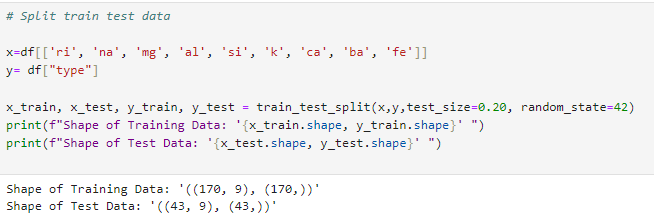
Visualization is an important activity during EDA. This enables us to see the high-level distribution of data and the patterns associated in it. Common types of visualizations are histograms, box plots, scatter plots, heat maps etc.



# Dense Neural Network Model

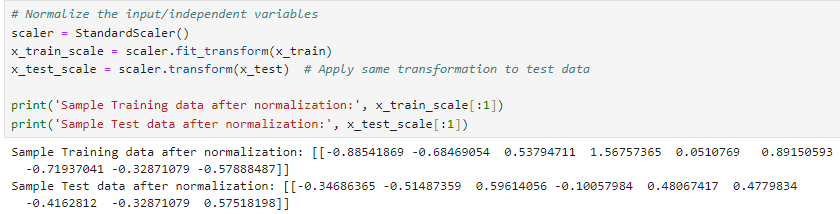
To build a working model, the first step is to load the relevant libraries and then split the actual dataset into training and testing. For this purpose, we use the ‘train\_test\_split’ function and split the total dataset into to 80/20 ratio with condition ‘random\_state’ 42.

After splitting the data into train and test, we can check the total number of data available for each of the process by using the ’shape’ parameter of data frame.



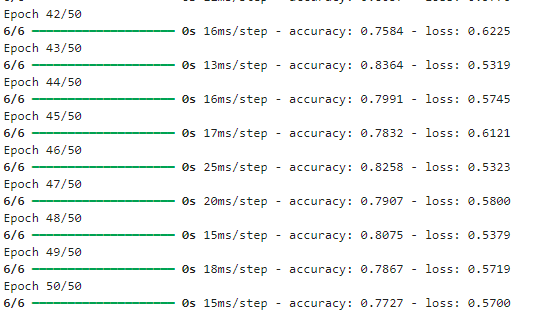
During the EDA, it was identified that the mean value of each of the input parameter was at different scale level. For the machine learning model to train effectively, it is important to scale the features so that they are within a specific range. For this purpose, we make use of the ‘StandardScaler’ function which is part of the ‘SKLEAR.PREPROCESSING’ library.

It must be clearly understood that using standard scaler the relationship between each data points are preserved but the mean, standard deviation and scale of data is altered. Also, the original data can be recovered using inverse transform method. Thereby, the dataset is not changed.



A simple model using dense neural network is built using randomly selected parameters - 2 hidden layer with 64 and 32 nodes, activation function ‘relu’ at hidden layer and ‘softmax’ for output layer. ‘Epochs’ value of 50 and batch size of 32 is taken for the training.

Using the ‘summary’ function, it is possible to see the number of parameters that will used during model training.  
It must be noted, that the above parameters are randomly assumed and not the best fit scenario.



Reading the accuracy rate at the last epoch, we can understand that the accuracy for training data is approx. 77% and the loss is at 0.57.

.

# Hyperparameter Tuning

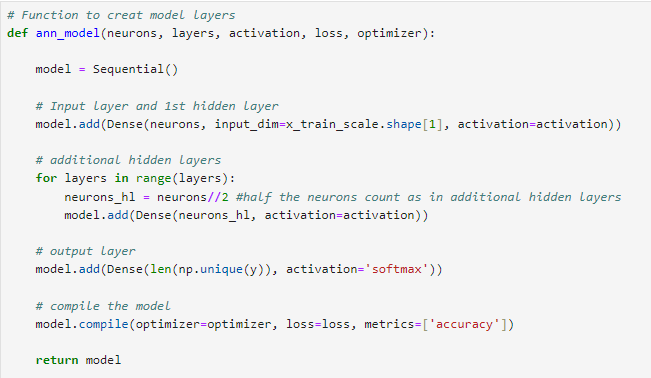
Hyperparameters are nothing but configurational parameters that are provided for algorithms. Each algorithm has their own configuration parameters that control the behaviour and performance of the model.

For example, some of the important configurational parameters pertaining to neural network algorithm are – ‘optimizer’, ‘loss’, ‘activation’, ‘layers’, ‘neurons’, ‘epochs’.

Each and all parameters have a specific setting that controls how the algorithm is executed. Thus, finding the optimal hyperparameters would help us achieve the best-performing model.

There are several techniques for choosing a model’s hyperparameters, including ‘RandomizedSearchCV’, ‘GridSearchCV’ etc. ‘GridSearchCV’ is widely recognized for its efficiency in tuning parameters and same as been taken for this study.

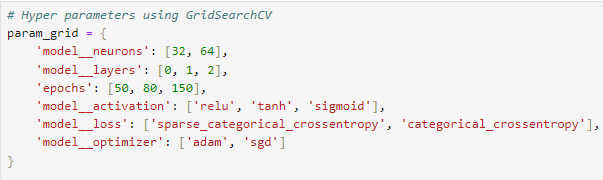
To evaluate the best parameter, a function namely ‘ANN\_MODEL’ has been built which takes the values of neurons, layers, activation, loss and optimizer and return the model object.



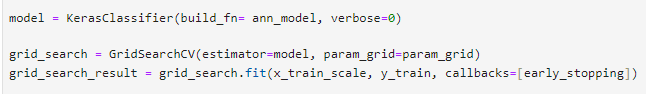
Since the process to find best parameters is too expensive and time consuming, we try to restrict the model training when there is no changes to the loss and accuracy after 5 iteration. To achieve this, early stopping function is used.



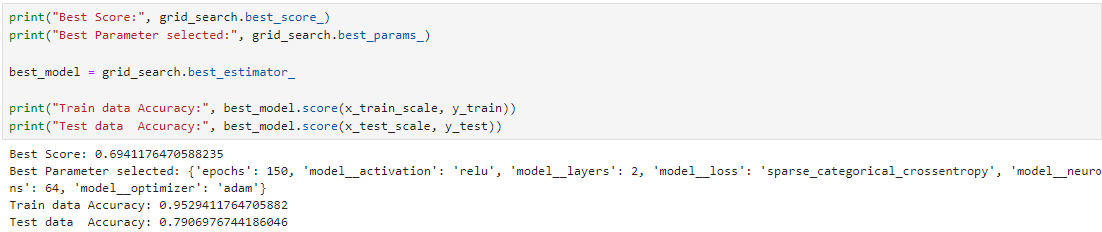
Since this process is about find the best parameter fit, we need to pass multiple values to each model parameter and therefore the values are maintained in a dictionary field as shown below.



To pass the parameter values maintained in the above-mentioned dictionary, we make use of the ‘KerasClassifier’ and ‘GridSearchCV’ function together as shown below.



Using this approach, it was identified that the best parameters for model training are as below.



## Cross Validation

Cross validation is a technique used to avoid overfitting scenario by training the model with multiple subsets of train-test data. To achieve this, k-folds is used which is a numeric value that is passed as input and based on which actual dataset is split into multiple smaller units and model is trained and tested with each unit. Using this approach, model is trained with all the possible data and tested multiple times. The average of all the result is taken as final measure.  
(Simplilearn)

# Classification Report

Classification report provides a detailed breakdown on the model evaluation. It shares key information like precision, recall, f1-score and support.

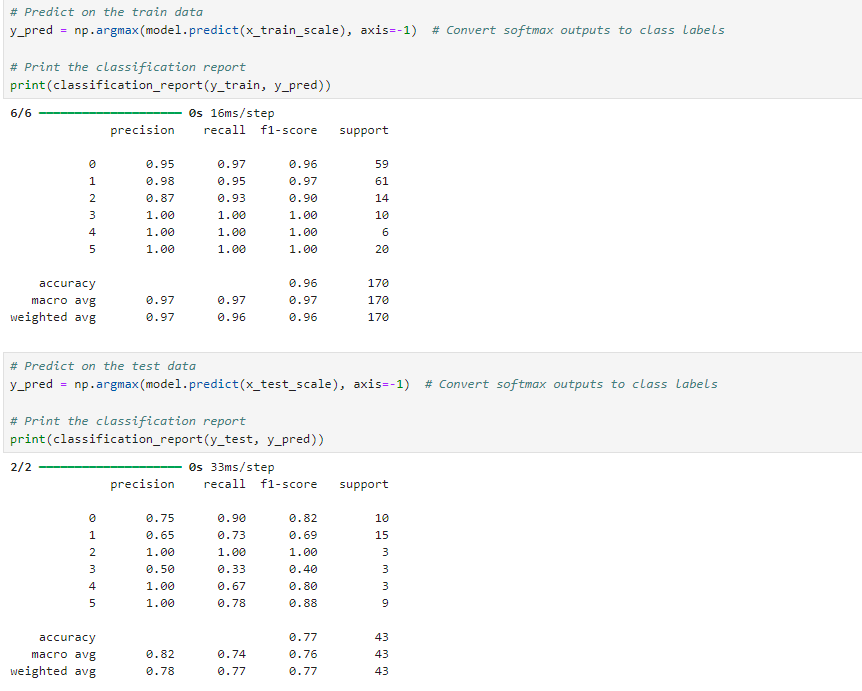
**Precision** is the proportion of correctly predicted positive instances from all the positive predicted instances. Low precision indicates many false positives.

**Recall** is nothing but true positive rate i.e. correctly predicted positive instances out of all actual positive instances. A lower value indicates many false negatives.

**F1-Score** is the actual precision value that tells the final accuracy rate.

**Support** is the total number of occurrences of each class in the respective model prediction.

Find below the classification report created for a model using the above recommended parameter from hyper parameter findings.

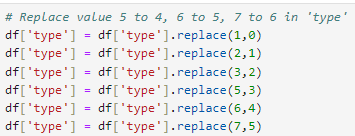


Reading the precision, recall, f1-score and support value from the above train data results, it is evident that the model prediction is 77% accurate with 100% accuracy for the specific glass type ‘2’.

# Conclusion

We must remember the changes done to class numbers in the data preparation stage. Therefore, we must understand that the class numbers in the result actually refers as following,

0 🡪 1, 1🡪 2, 2🡪 3, 3🡪 5, 4🡪 6, 5🡪 7.



From the classification report, it is evident that prediction is 100 accurate for class 3 (vehicle\_windows\_float\_processed), 88% accurate for class 7 (headlamps), 82% accurate for class 1 (building\_windows\_float\_processed), 80% accurate for class 6 (tableware), 69% accurate for class 2 (building\_windows\_non\_float\_processed), 40% accurate for class 5 (containers).

# References

(Melanie)

Melanie. “Dense Neural Networks: Understanding Their Structure and Function.” *Data Science Courses | DataScientest*, 5 Mar. 2024, datascientest.com/en/dense-neural-networks-understanding-their-structure-and-function.

Available at:

<https://datascientest.com/en/dense-neural-networks-understanding-their-structure-and-function>

(Corpnce)

Corpnce. “Corpnce.” *Corpnce*, 22 Dec. 2023, www.corpnce.com/5541-2/.

Available at:

https://www.corpnce.com/5541-2/

(Tariq A)

Tariq A. “Exploratory Data Analysis (EDA) Is a Critical Step in the Machine Learning Process. It Involves Examining Datasets to Uncover Patterns, Spot Anomalies, and Test Hypotheses before Moving on to Model Building.” *Linkedin.com*, 21 Aug. 2024, www.linkedin.com/pulse/comprehensive-guide-exploratory-data-analysis-eda-machine-tariq-ali-lnwmc/. Accessed 15 Dec. 2024.

Available at:

<https://www.linkedin.com/pulse/comprehensive-guide-exploratory-data-analysis-eda-machine-tariq-ali-lnwmc/>

(Simplilearn)

Simplilearn. “The Ultimate Guide to Cross Validation in Machine Learning for 2021.” *Simplilearn.com*, 18 Mar. 2024, www.simplilearn.com/tutorials/machine-learning-tutorial/cross-validation.

Available at: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/cross-validation>

# GitHub Link

<https://github.com/santhosh-sba24100/CA---Python_Programming>