**Building of ANN for Wine\_Quality\_Check:**

* After performing the necessary data cleaning steps, such as removing missing values, outliers, and duplicates, I imported the resulting CSV file into a pandas data frame for further analysis.
* This data frame contains the relevant variables and observations that I need to explore and manipulate for my research project.
* One of the important steps in data analysis is to ensure that the data frame has the correct data types for each column. This can affect the accuracy and efficiency of the subsequent operations and calculations.
* Rechecking the data types of a data frame is a good practice to avoid errors and ensure the quality of the data analysis.
* Identifying the Input and output features.
* To identify the input and output variables in the data frame, we need to have a clear research question or hypothesis that specifies what we are interested in studying.
* One of the common steps in Machine learning and Deep learning is to split the data into train and test sets.
* This is done to evaluate the performance of a model on unseen data and to avoid overfitting.
* The train set is used to train the model, while the test set is used to measure how well the model generalizes to new data.
* The ratio of the train and test sets also varies depending on the problem and the data availability.
* A common rule of thumb is to use 80% of the data for training and 20% for testing, but this may not always be optimal. The goal is to find a balance between having enough data to train a robust model and having enough data to test its accuracy and generalizability
* One of the important steps in data analysis is checking for imbalances in the data. An imbalance occurs when one or more classes of the target variable have significantly more or less observations than the others. This can affect performance and accuracy.
* One of the common tasks in data preprocessing is dividing the columns of a dataset into numerical and categorical ones.
* This can help to apply different methods of analysis, transformation, and visualization for each type of column.
* One hot encoding is a technique for transforming categorical variables into numerical ones.
* It works by creating a new column for each unique value in the categorical column and assigning a 1 or 0 to indicate whether that value is present or not.
* Standardization is a common preprocessing technique of machine learning and deep learning.
* It transforms the features of the data to have zero mean and unit variance, which can improve the performance and convergence of the models.
* Apply the same transformation to the test data using the mean and standard deviation from the training data. This ensures that the data is consistent and comparable across different datasets.
* An artificial neural network (ANN) is a computational model that mimics the structure and function of biological neural networks.
* An ANN consists of a collection of interconnected units called artificial neurons, which process information by applying activation functions and weights to their inputs.
* The weights determine how strongly each input affects the output of a neuron, and the activation functions determine the output value of a neuron given its weighted inputs.
* The logical structure of an ANN can be divided into three main components: the input layer, the hidden layers, and the output layer.
* The input layer receives external data and passes it to the hidden layers. The hidden layers perform the main computation and learning tasks of the network, by applying various transformations and algorithms to the input data. The output layer produces the final results or predictions of the network, based on the outputs of the hidden layers.
* The number and size of the hidden layers, as well as the choice of activation functions and weights, depend on the type and complexity of the problem that the ANN is designed to solve.
* Different types of ANNs have different logical structures, such as feedforward networks, recurrent networks, convolutional networks, etc. The logical structure of an ANN determines its capabilities and limitations, as well as its performance and efficiency.
* Compiling of model in ANN is the process of configuring the learning algorithm and specifying the loss function, the optimizer, and the metrics to evaluate the model performance.
* Keras Tuner is a library that helps you find the best hyperparameters for your deep-learning models. Hyperparameters are the variables that control the architecture and training of a model, such as the number of layers, the activation functions, the learning rate, etc. Finding the optimal hyperparameters can improve the performance and efficiency of your models.

* You can use Keras Tuner with any Keras model, but you need to define a function that builds and compiles the model. This function takes a single argument, which is an instance of the HyperParameters class.

* You can use this object to sample hyperparameters from predefined distributions, such as `hp.Int`, `hp.Float`, or `hp.Choice`. For example, you can define a function that creates a convolutional neural network with a variable number of layers, filters, kernel size, and activation function.

* One way to optimize the hyperparameters of a deep learning model is to use the Keras tuner Random Search. This method randomly samples the values of the hyperparameters from a predefined distribution and evaluates the model performance on a validation set. The best hyperparameter combination is then selected based on the validation score. Keras tuner Random Search is easy to implement and can explore a large search space efficiently.

* The results\_summary() method of Keras Tuner is a useful tool to display the best hyperparameters found during the tuning process. It shows the values of the hyperparameters that achieved the highest score on the validation set, as well as the score itself.

* To use this method, you need to have a tuner object that has performed the search() method on a dataset. Then, you can simply call tuner.results\_summary() to see the summary.

* The search\_space\_summary() method of the Keras Tuner is a useful tool to inspect the search space of a hyperparameter tuner. It prints a summary of the hyperparameters and their possible values, as well as the conditional dependencies between them.

* The summary can help to understand the range and diversity of the configurations that the tuner will explore during the hyperparameter optimization process.

* To use the search\_space\_summary() method, you need to pass it a tuner object that has been initialized with a model-building function and a hyperparameters object.

* Keras Tuner provides a method called get\_best\_models, which returns a list of trained model instances sorted by their performance on the validation data. To get the best model from this list, you can simply use the index 0, as in get\_best\_models(num\_models=1)[0]. This will return the model with the highest validation accuracy or the lowest validation loss, depending on how you defined your objective function.

* One of the useful features of the Keras Tuner is that it allows you to inspect the best model architecture and parameters after the hyperparameter tuning process. To do this, you can use the model.summary() method on the tuner object, which will print out a summary of the model layers, output shapes, and number of parameters. This can help you understand how the model works and what are the most important hyperparameters for your problem.

* One of the benefits of using Keras Tuner is that it allows you to easily access the best hyperparameters found by the search. You can do this by calling the get\_best\_hyperparameters() method on the tuner object, which returns a list of HyperParameters objects. The first element of this list corresponds to the best hyperparameters, and you can access its values as a dictionary.

* To start the tuning process, you need to call the tuner's search method, which takes the same arguments as model.fit(), such as the training data, the validation data, the number of epochs, and the batch size.

* The tuner will then train and evaluate multiple models using different sets of hyperparameters, and store the results in a directory specified by the project\_name argument.

* Plotting the graph for mean squared error (mse) and validation mean squared error (val\_mse) on training and validation data in artificial neural network (ANN).