Fossil Age Prediction Using ANN Model

Problem Statement:

The aim is to predict the age of fossils based on geological, chemical, and physical attributes using Artificial Neural Network Model.

Dataset Overview:

The dataset consists of 4,398 fossil samples with 13 features:

uranium_lead_ratio: Ratio of uranium to lead isotopes in the fossil sample.

carbon_14_ratio: Ratio of carbon-14 isotopes present in the fossil sample.

radioactive_decay_series: Measurement of the decay series from parent to daughter isotopes.

stratigraphic_layer_depth: Depth of the fossil within the stratigraphic layer, in meters.

isotopic_composition: Proportion of different isotopes within the fossil sample.

fossil_size: Size of the fossil, in centimeters.

fossil_weight: Weight of the fossil, in grams.

geological_period: Geological period during which the fossil was formed.

surrounding_rock_type: Type of rock surrounding the fossil.

paleomagnetic_data: Paleomagnetic orientation data of the fossil site.

stratigraphic_position: Position of the fossil within the stratigraphic column.

age: Calculated age of the fossil based on various features, in years.

Data Pre-Processing:

Essential libraries such as NumPy and Pandas for data manipulation, and Matplotlib and Seaborn for visualization, were imported. The dataset was copied to ensure original data

integrity, and checks for data size, shape, and correctness of column names and types were conducted. The 'inclusion_of_other_fossils' column was converted to an integer format for ease of analysis. No null values or duplicates were found. The dataset was then split into categorical and numeric variables for detailed univariate analysis.

Univariate Analysis:

Categorical Data Analysis:

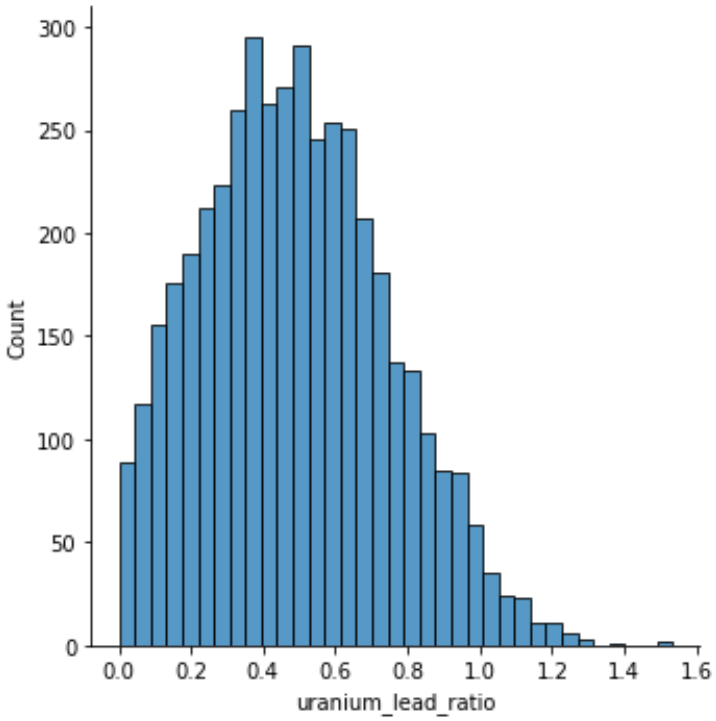
The initial phase focused on examining categorical data by iterating through each categorical column to identify subclasses and determine their distribution within the dataset. This process provided insights into the diversity and frequency of subclasses present, essential for understanding their impact on the target variable.

```
geological_period
Cambrian          882
Triassic          676
Cretaceous        601
Devonian          498
Jurassic          490
Paleogene         405
Permian           365
Neogene           311
Ordovician        100
Carboniferous      52
Silurian           18
Name: geological_period, dtype: int64
paleomagnetic_data
Normal polarity      3160
Reversed polarity    1238
Name: paleomagnetic_data, dtype: int64
surrounding_rock_type
Sandstone        1497
Limestone        1166
Shale            1144
Conglomerate      591
Name: surrounding_rock_type, dtype: int64
stratigraphic_position
Bottom    2667
Middle    1267
Top        464
Name: stratigraphic_position, dtype: int64
```

Numeric Data Analysis:

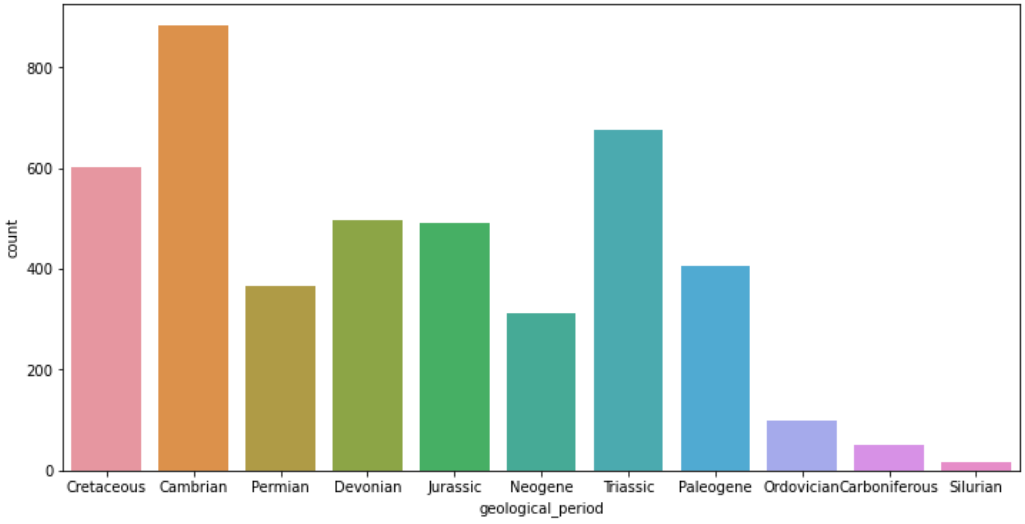
Attention then shifted to numeric columns. Box plots were generated to assess the distribution and presence of outliers in each numeric attribute. This step revealed outliers in certain columns, highlighting the need for outlier treatment to enhance the reliability of

subsequent analyses and model performance. We build some distribution plots as well to see how each variable distributed.one of the example given below.



Count Plot Analysis:

Count plots were generated for each categorical column to visualize the spread and frequency of subclasses. This facilitated a deeper understanding of how categorical attributes contribute to the dataset's composition and variability.one of the example given below.



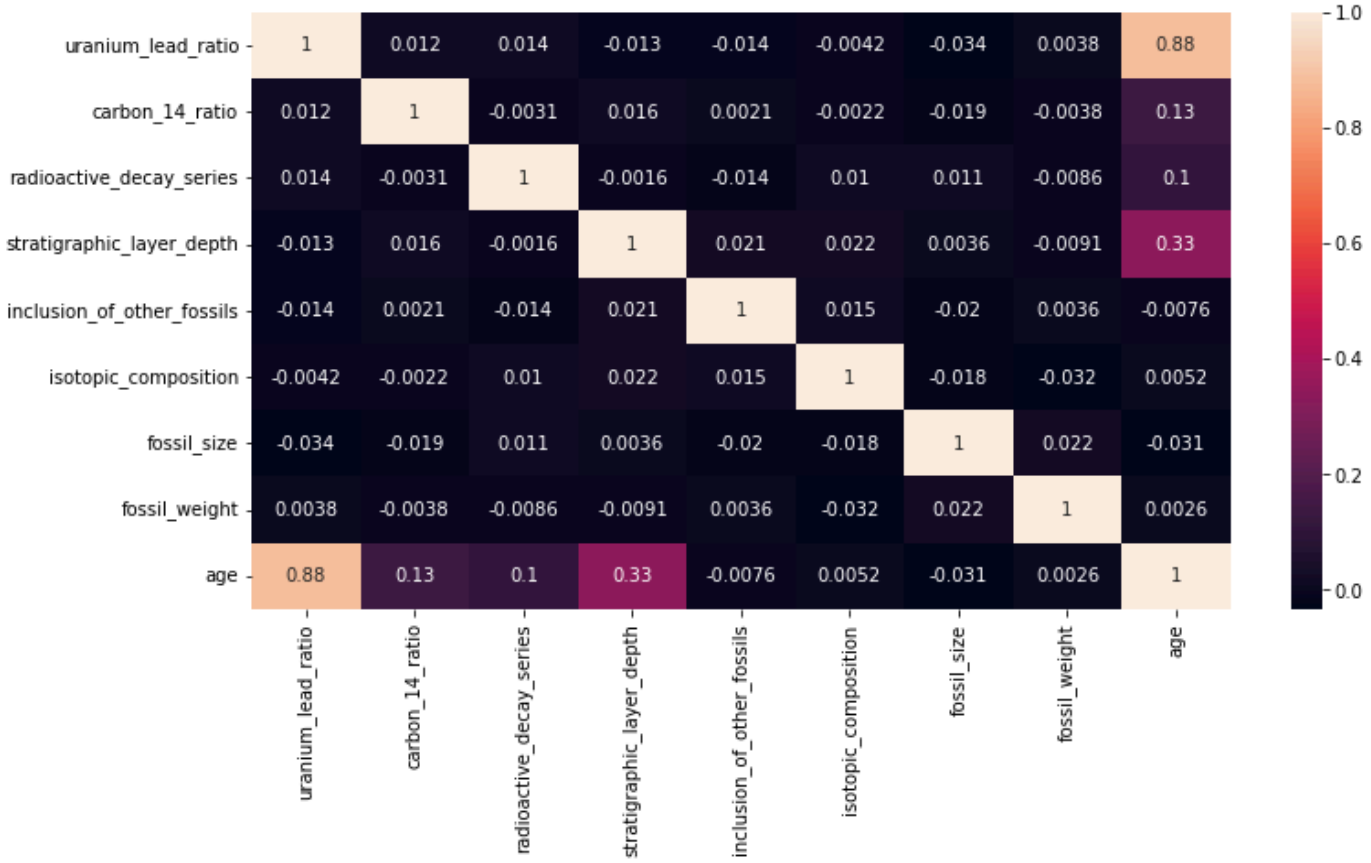
Comprehensive Analysis:

Each column's characteristics, spread, and value distributions were thoroughly evaluated within the entire dataset context. Detailed visual representations are best explored in the accompanying notebook for clarity and precision.

Bi-Variate Analysis:

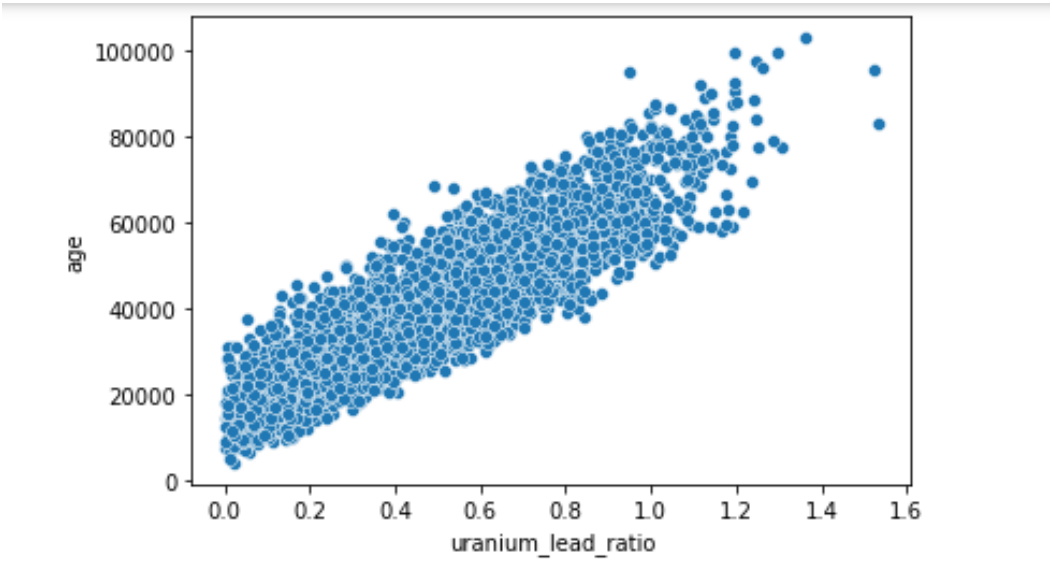
Heatmap of Correlation:

A heatmap was generated to visualize correlations between variables in the dataset, focusing on their impact on the target variable 'age'. The analysis revealed no significant multicollinearity issues among the predictors, with the 'uranium_lead_ratio' exhibiting a notably strong positive correlation with 'age', suggesting its pivotal role in predicting fossil ages.



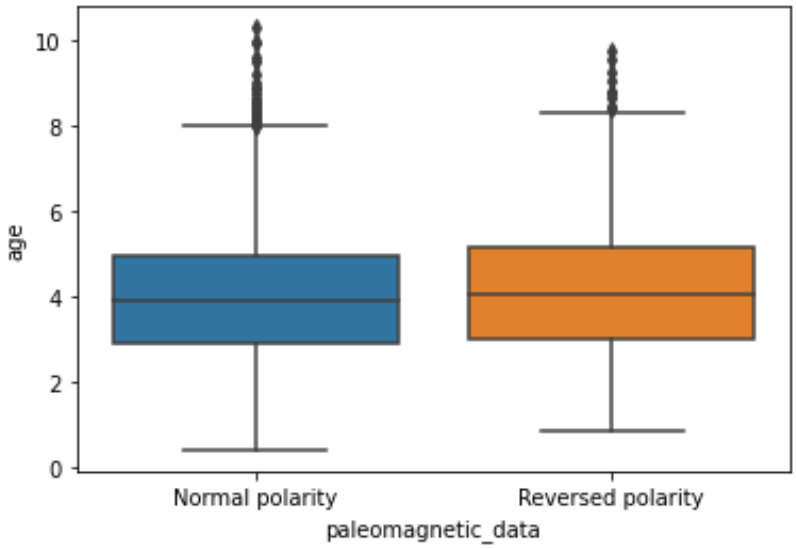
Scatter Plots:

Scatter plots were created for each numeric variable plotted against the target variable 'age'. These plots identified patterns and relationships, particularly highlighting the 'uranium_lead_ratio' as having a positive correlation with age.



Box Plots for Categorical Variables:

Box plots illustrated how different subclasses within each categorical attribute relate to fossil age, providing insights into their contributions to the predictive model.one of the plots given below.



Outlier Analysis and Treatment:

Outlier analysis involved checking if outliers in a particular column have any relation with other columns. Pair plots and value counts for categorical columns were used for this

purpose. It was found that outliers in numeric columns are related to other variables; hence, outlier treatment was not performed.

Feature Scaling:

Given that the numeric variables do not follow a normal distribution, Min-Max Scaling was applied instead of Standard Scaling. This approach was chosen based on the range of values for each variable.

Model Building:

Two Artificial Neural Network (ANN) models were built: one without scaled values and one with scaled values. The first ANN model, The Sequential model is initialized and the first layer is added using the Dense function with 32 neurons, an input dimension of 12, ReLU activation function, and L2 regularization with a penalty of 0.01. L2 regularization is used to prevent overfitting by adding a penalty term to the loss function proportional to the square of the magnitude of the coefficients. Dropout with a rate of 0.2 is added after the first layer to randomly drop out 20% of the neurons during training, which also helps prevent overfitting.

The second layer is added using another Dense function with 10 neurons, ReLU activation, and L2 regularization with a penalty of 0.01. Dropout with a rate of 0.2 is added after the second layer as well.The final layer is added using a Dense function with one neuron and a linear activation function, which is appropriate for regression problems.The model is then compiled using the Adam optimizer with a learning rate of 0.001 and mean squared error loss function, which is commonly used for regression problems. As we look at the train loss and test loss we found out that the model we built is underfitting So let’s build a another model with cross validation to overcome underfitting by using scaled values.

```
28/28 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 2.1912
Test Loss: 2.1791138648986816
```

To address this, a second ANN model was built using cross-validation with scaled values. It first concatenates the scaled features and one-hot encoded categorical variables to create the feature matrix X1. The target variable y1 is also defined. The KFold cross-validator is initialized with 5 splits, shuffling the data and setting a random state. The code then iterates through each fold, splitting the data into training and validation sets using the current fold's indices. A neural network model is then defined with two hidden layers of 64 and 32 neurons respectively, using ReLU activation functions. The output layer has one neuron with a linear

activation function for regression. The model is compiled with the Adam optimizer and mean absolute percentage error (MAPE) loss function. The model is then trained on the current fold's training data for 10 epochs with a batch size of 32, using the validation data to evaluate performance during training. This process is repeated for each fold, and the final performance is averaged over all folds. This approach helps to reduce overfitting and obtain a more reliable estimate of the model's performance on unseen data. It’s somewhat deal with our underfitting problem but we got approximately equal loss in training and testing loss. Better than our Old Model.

```
28/28 [==============================] - 0s 1ms/step - loss: 0.0519
Validation Loss: 0.05193961039185524
```