Report on Prediction of Fossil Age using Regression Models

1. **Data Preparation.**
   To import the dataset, we used the 'read_csv' function from pandas. Before importing the Dataset, we had looked for irrelevant columns but there aren't any present, so we move on to Data Encoding. Where we convert categorical values to numerical values so that machine learning algorithms can understand the data. For the features 'inclusion_of_other_fossils', and 'paleomagnetic_data' we used map function as they contain binary values as well for 'stratigraphic_position' and 'geological_period' as they can be ranked. We have one more categorical column 'surrounding_rock_type' but we will use Target Encoding to encode that column as it cannot be ranked. But first, we split the data between input and output columns as Target Encoding is based on output column.
   The output column for our dataset is 'Age' as we are predicting the Age of the fossil. Next, we look for dependent (high correlation) columns in our data by using 'heatmap'. Our dataset does not have any high correlation between input columns, so we move on to last step in Data preparation which is Data Scaling. Where we apply 'StandardScalar' to the input columns to ensure that all features contribute equally to the model's performance, regardless of their original scale. This method can be called as Data Standardizing or Data Normalizing.

2. **Impact of L1, L2, and elastic net regularization on linear regression coefficients, performance, and interpretability.**
   L1, L2 and Elastic Net are different penalties or regularizations we apply to linear regression models. Focusing on coefficients, L1 gives absolute value of size of the coefficients (denoted by modulus in formula), due to which some coefficients can be zero. The effect of which will be better for feature selection as it focuses on a part of features. L2 adds a penalty of square value of size of coefficients which decreases the value of coefficients but not till zero. Elastic Net is a combination of both L1 and L2, so it sets some coefficients to zero and decreases some values as well.
   As for Performance, L1 focuses on only important features which leads to a smaller number of irrelevant features, increasing performance. L2 is better in handling collinearity as it distributes the impact among correlated features. Elastic net again balances both as it is a combination of L1 and L2.
   L1 is most interpretable as it turns the irrelevant features in to zero values. Leaving few important features to interpret. L2 does not set coefficients to zero values and distributes importance among all features, making it less

interpretable. Elastic Net is more interpretable than L2 as some coefficients can be zero but still less clear than a L1 model.

But in our case all models give similar coefficient values and interpretability. The only difference is observed in performance as L1 and L2 takes similar time to execute, whereas Elastic Net takes 5x more than above mentioned two models.

3. **Impact of L2 regularization on support vector regression performance and interpretability.**

   The First impact we notice in application of L2 regularization (C) in SVR is the best score i.e. R-squared error (r2) we get as a result. When we remove the regularization parameter (C) from the model we get ‘r2’ as ‘0.32’ (on the scale of 0-1) whereas when we give a list of ‘C’ values as a parameter we get ‘0.98’ r2 value, which is a significant difference when performance of a model is considered.

   As for interpretability of both the models, model without the application of L2 is more interpretable as it gives the best kernel as linear and thus coefficients can be extracted for interpretability. As compared to L2 regularization SVR, where coefficients aren’t available for interpretation because we get ‘rbf’ as best kernel. With the exception, when we keep smaller ‘C’ where we get linear as a best model but with smaller r2. One more difference between both models is the execution time, model with L2 regularization takes 3 minutes to execute on the other hand it takes 20 seconds without the regularization parameter.

4. **If you were to implement random forest regression, then its comparative performance and interpretability with respect to regularized linear regression and regularized support vector regression models.**

   First, we’ll compare Random Forest Regression (RFR) with Regularized Linear Regression (LR). Judging on performance, regularized LR models give good results i.e. ‘r2’ as compared to RFR with the difference of ~0.3. This might be due the fact that our data is linear. As for interpretability, RFR is less interpretable than LR in nature but ‘feature.importances_’ feature helps in understanding which are the important features contributing to the score, whereas LR tells exactly how much a feature impacts the model with use of Coefficients. In our case, both LR and RFR take similar time to execute.

   Comparing Regularized Support Vector regressor (SVR) and RFR, SVR gives 0.4 more ‘r2’ value than RFR, this again might be since data our data is linear and SVR’s capability of handling both linear and non-linear with the help of kernels, whereas Random Forest Regressor is best for non-linear data. One more reason for better performance of SVR than RFR might be due to our data is small sized. Theoretically, SVR is considered more interpretable than RFR but since we get best kernel as gaussian (rbf) instead of ‘linear’ we are unable to interpret the results as coefficients values aren’t available. As for execution times RFR takes 3x less time than regularized SVR as SVR has more type and number of parameters.

5. **Performing a prediction for one of your models using new data.**
   We created a l1fossilmodel.pkl file for our model with the use of ‘joblib’ library. The model we selected for further testing is ‘L1’ as the results were similar in all the linear regression models, but L1 needed a smaller number of maximum iterations as compared to others.
   Now, referring to ‘Predicting Fossil Age using PKL.ipynb’ present in the zip file. First, we import the test dataset (also present in zip file) and apply the same data preparation steps as earlier. When our new data is ready, we import our ‘.pkl’ file saved earlier. Then we ‘predict’ the output column ‘age’ by giving it input features. For comparison purposes, we insert the predicted output column ‘predicted age’ in our original test data set.