Before You Begin:

- Make sure required libraries are installed. These libraries are listed at the beginning of the code.
- Code is based on ‘Collected Data - Data Points.csv’. It is coded as a relative path. Make sure .csv is in project file (Ex: C:\Users\Name\PycharmProjects\PythonProject1)

Notes:

- I am still actively collecting data. Variables are still highly correlated because of a small dataset. Once there is a sufficient amount of data, feature engineering will be more accurately conducted. Accuracy scores are low because of the small dataset. See attached spreadsheet for all data collected so far and sources that data will be collected from.
- Code is still a work in progress. I still need to handle missing values, outliers, and duplicates once there is a sufficient amount of data points. Additionally, Optuna will be finetuned to more accurately fit data. Finally, a post-analysis will be carried out to better understand each feature’s contribution to the final output to better understand sulfuric acid’s effect on geopolymerization.
- At the moment, I have dropped the variable *Mass Change (%)* from the model because of insufficient data.

Code Output:

1. Prints ‘Collected Data - Data Points.csv’ as pandas dataframe

2. Prints count, mean, standard deviation, minimum, 25th quartile, median, 75th quartile, and maximum values of each variable
3. Pop-up window of histograms of values of each variable. Close to continue.
4. Pop-up window of boxplots of values of each variable. Close to continue.
5. Pop-up window of heatmap showing correlations between each variable (*Slag (kg/m3)* variable is missing because of lack of variation in data; *Curing Time (hr)* x *Mass Change (%)* is missing because of lack of data). Heatmap will be used to determine variables used in the final model to prevent multicollinearity. Close to continue.
6. Optuna optimizes hyperparameters used in the XGBoost model. Prints R^2 values for each fold for each trial and prints and returns the mean of these R^2 values for each trial.
7. Prints hyperparameter values of the trial that resulted in the highest R^2 value.
8. XGBoost is trained with these hyperparameter values. Outputs verbose of XGBRegressor training (evaluation metric = RMSE)
9. Prints best iteration (the specific boosting round with the best performance [lowest RMSE])
10. Prints evaluation metric value of the best iteration
11. This model is then evaluated on its predictions of the validation dataset. RMSE, MAE, MSE, and R^2 values are printed.
12. GUI will pop up. Inputting values and pressing the button will output a predicted compressive strength (MPa). Close to end code.